DOCUMENT RESUME

ED 315 925 EC 222 028

AUTHOR	Russell, Susan Jo; And Others
TITLE	Beyond Drill and Practice: Expanding the Computer Mainstream.
INSTITUTION	Council for Exceptional Children, Reston, Va.
SPONS AGENCY	Special Education Programs (ED/OSERS), Washington, DC.
REPORT NO	ISBN-0-86586-190-0
PUB DATE	89
GRANT	G008430071
NOTE	130p.
AVAILABLE FROM	Council for Exceptional Children, Publication Sales, 1920 Association Dr., Reston, VA 22091 ($18.00, $14.40 members; stock no. P333).
PUB TYPE	Guides - Non-Classroom Use (055)
EDRS PRICE	MF01/PC06 Plus Postage.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Software; Critical Thinking; *Disabilities; Elementary Secondary Education; Learning Strategies; Mathematics Instruction; Problem Solving; Reading Instruction; *Skill Development; Special Education; Teacher Student Relationship; *Teaching Methods; Word Processing

ABSTRACT

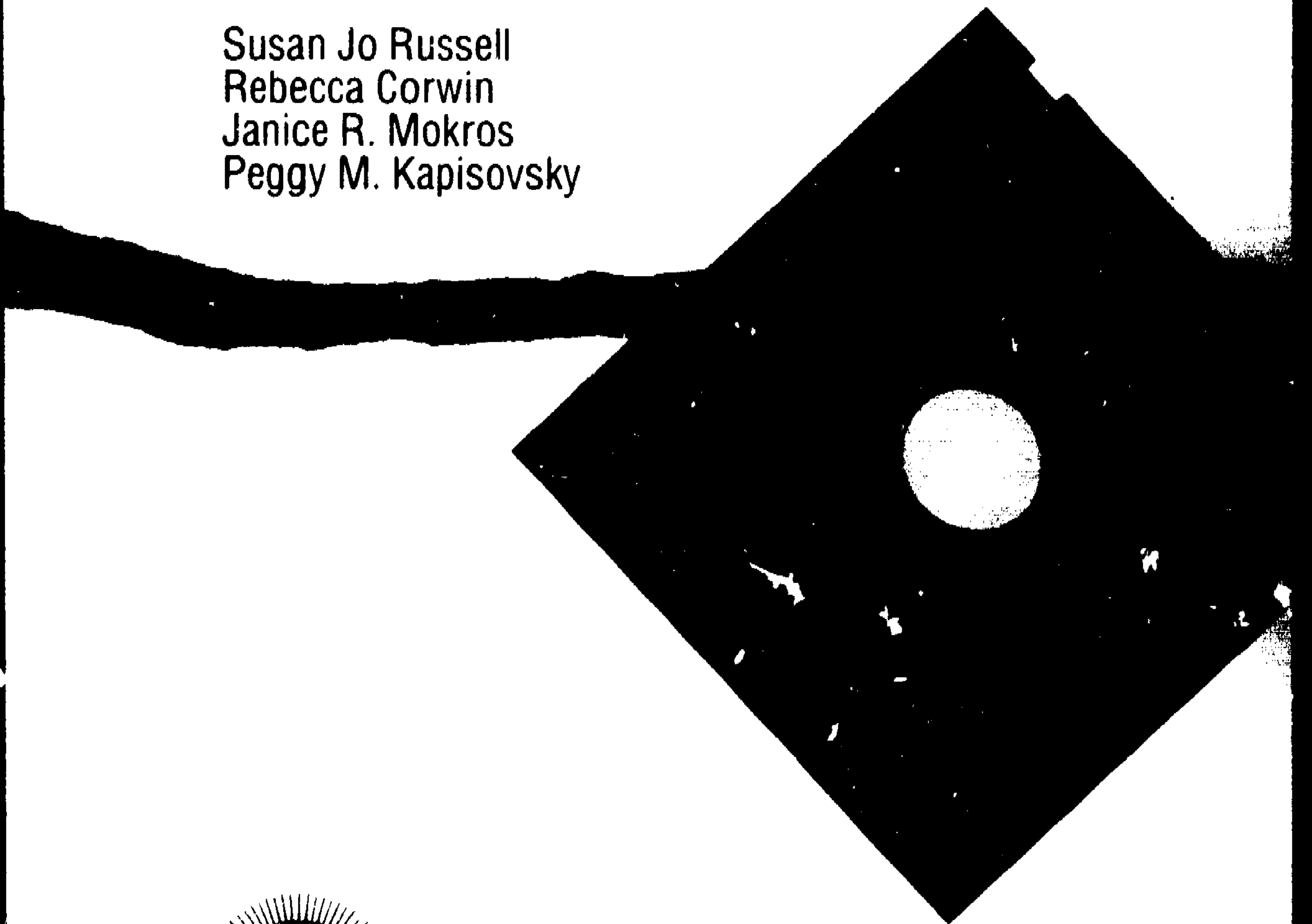
This book describes the practice of using learner-centered software in special education. Following an introductory chapter, Chapter 2 provides a discussion of software that goes beyond drill and practice and at the same time fits easily into the existing subject areas of reading/language arts and mathematics. Chapters 3 and 4 discuss using word processing software to teach writing and using software to develop problem-solving and critical thinking skills. Numerous illustrations of effective ways that teachers have used the software are included. In Chapter 5, the topic of using software to help students learn how to be better learners is addressed. Chapters 6 and 7 deal with the interaction between student, learning environment, and software--all of which is facilitated by the teacher. Chapter 6 focuses on the multiple roles of the teacher as introducer, technical advisor, arranger, visitor, silent partner, booster, mentor, and learner. The final section of Chapter 7 lists alphabetically all software identified in the book, with publisher, address, and computer type. (JDD)

Beyond Drill and Practice:

Beyond Drill and Practice:

Expanding the Computer Mainstream

Susan Jo Russell
Rebecca Corwin
Janice R. Mokros
Peggy M. Kapisovsky



CEC

Published by The Council for Exceptional Children

Library of Congress Cataloging-in-Publication Data

Beyond drill and practice: expanding the computer mainstream / Susan Jo Russell . . . [et al.].
p. cm.
ISBN 0-86586-190-0
1. Special education—United States—Computer-assisted instruction. I. Russell, Susan Jo.
LC3969.5.B49 1989
371.9'043—dc20 89-15913
CIP

This book was supported in part by Grant No. GOO 8430071 from the Office of Special Education Programs, U.S. Department of Education.

Stock No. P333 Price $18.00

Printed in the United States of America.
10 9 8 7 6 5 4 3 2 1

Contents

Dedication

This book is dedicated to the memory of Betty Church and all special needs teachers willing to explore and extend new ideas and new technologies.

Acknowledgments

This book is largely based on the work of many special education teachers. Substantial contributions to this book were made by:

Temple Ary
Michael Brandmeyer
Mary Briggs
Betty Church
Marguerite Nelson Creskey
Tom Plati
Charlotte Reid
Jan Schraitle
Donna Simone
Dorothy Spahr
Steve Spencer
Dawna Traversi
Steve Voiles
Linda Ware

Special thanks for supporting and contributing to the Beyond Drill and Practice project also go to June Foster, Director of the Special Needs Center at Technical Education Research Centers; to Jane Hauser of the Office of Special Education Programs in the U.S. Department of Education; to Joan Thormann, Division of Special Education, Massachusetts Department of Education; and to LeAnn Davis and Elisabeth Roberts of Technical Education Research Centers for editorial and production assistance.

About the Authors

Susan Jo Russell began her educational work as a classroom teacher and staff developer. As part of the research staff at M.I.T.'s Laboratory of Computer Science, she worked closely with staff at the Cotting School for Handicapped Children to develop educational programs through use of the computer for students with special physical and learning needs. Since coming to Technical Education Research Centers (TERC) in 1983, Dr. Russell has directed research and development projects in computer education, special education, and mathematics education, including the project on which this book is based. Currently, she directs two projects funded by the National Science Foundation, which focus on mathematics learning and teaching in the elementary grades, and she is part of a research team conducting a 3-year study of the integration of technology into the education of middle school special needs learners, funded by the U.S. Department of Education.

Rebecca B. Corwin is Associate Professor of Education at Lesley College Graduate School in Cambridge, Massachusetts, and senior associate to the Mathematics Center at TERC. A former elementary school teacher, she currently teaches graduate courses to preservice and inservice teachers in mathematics education and in using computers with a variety of students, including those with special needs. Her research focuses on helping teachers and students make sense of what they do on a daily basis. Much of her work in mathematics education and computer use focuses on learned helplessness and anxiety as those attitudes prevent learners from realizing their full potential.

Janice R. Mokros is a developmental psychologist who co-directs the Mathematics Center at TERC. She specializes in educational research and evaluation and has conducted numerous studies on the impact of mathematics and science curricula developed at TERC.

Peggy M. Kapisovsky was project associate on Beyond Drill and Practice, the project on which this book is based. As head of Microcomputers in Special Education/Special Interest Group (micro-SIG), she has served as a resource to many teachers in their explorations of learner-centered software. She is currently Communications Director at TERC.

Preface

One of the frustrations of teaching as an occupation and profession is its extensive individual and collective amnesia, through which the best creations of its practitioners are lost to both contemporary and future peers (Shulman, 1987).

Teachers' stories about their students' learning are critical sources of information about what works. Teachers engaged in active practice and in careful reflection on that practice have much to say but little time or opportunity to say it. This book is an attempt to capture what experienced educators know from their own work in using computers with students who have special learning needs.

Four years ago, when the project out of which this book grew began, there was little knowledge about computer use in special education. Computer use in special education was then, and in large part still remains, a grass-roots movement. Those who had information about special education students' use of computers were those practitioners who were taking risks with new ideas in their own classrooms. While there is now a growing body of research which will help practitioners think about integrating computers into their curriculum, research will never provide all the answers, just as it will never provide all the answers about reading or mathematics. The decision about what to try with one particular, individual student at one particular moment in his or her educational history will continue to be a judgment call based on the teacher's knowledge of many factors, including that student's demeanor when he or she walks through the classroom door.

This book is a compilation of such judgments by experienced teachers, teacher trainers, curriculum developers, and researchers. In our view, knowledge generated by practitioners must be weighed equally in the balance with knowledge generated by those outside the classroom. It is the goal of this book to make knowledge from practice available to other practitioners, and to set this information in a coherent context of knowledge about teaching and learning.

The gathering of the many experiences of individual teachers was largely supported by a grant to Technical Education Research Centers (TERC) from the Federal Office of Special Education and Rehabilitative Services for a project called "Beyond Drill and Practice: Microcomputers in Special Education." This funding made possible both a national search for promising practices and the development of a regional network of computer-using special educators, which continues to be coordinated through TERC's Special Interest Group.

Many special education teachers have contributed their experiences to this book. Linda Ware, Steve Spencer, Jan Schraitle, and Steve Voiles, who contributed heavily to the chapter on problem solving, were some of the first teachers in the country to explore the use of problem-solving software with special education students. Temple Ary's creative application of *Logo* in her mathematics classes for learning disabled students has become a model for many other special-needs teachers. The work of Betty Church and that of Donna Simone provide evidence about how resource-room students can become proud and confident authors through extensive use of the word processor. The list could go on and on.

When our research began in 1984, many of these teachers were just beginning to try a few activities with their students. In the years of work on this book, some of them have taken on various leadership roles in the field, including training, writing, and research. It is the grass-roots work of teachers like these that makes special education a vital and growing profession.

This book is dedicated to these teachers and to all the risk takers in special education who refuse to be defined and limited by what is expected, and who continue to believe that change is possible.

Susan Jo Russell
Cambridge, Massachusetts
December 1988

REFERENCE

Shulman. L. S. (1987). Knowledge and teaching: Foundations of the new reform. In M. Okazawa-Rey, J Anderson, & R. Traver (Eds.), *Teaching teachers of teacher education* (pp. 313–334). Cambridge, MA: Harvard Educational Review.

1. Charting a New Course: Drill, Practice, and Beyond

Is the computer an effective tool in special education? Special education administrators and teachers are asking this question. Yet such a general question about computer use is, at best, unproductive and, at worst, misleading, moving us away from a careful, differentiated look at the computer's potential. If asked whether *books* have an impact on special-needs students, educators and researchers would respond without hesitation: "It depends. To teach what? For whom? In what teaching context?" and, of course, "But what book do you mean?" Persistent myths prevent us from asking similar questions about computer software. For example, we may believe that

- The computer should do it all; once the student is engaged with the computer, the teacher is no longer needed.
- The computer is doing it right; the design of computer software is based on sound pedagogical principles and sound content.
- A good piece of computer software should present no difficulties; if we, or our students, have any trouble with the directions or mechanics of the software, it should be discarded.
- A good piece of software is good for all students.

Our good sense and educational experience tells us these statements are not true, although sometimes we act as if they are. As computers become more familiar in special education, educators and researchers have started asking better questions. For which students, in what context, under what conditions, with what teacher interventions, and for what purpose is this piece of software appropriate? Choosing the right software for special education students demands as much care and attention as choosing any other course materials. Similarly, supporting good software with an appropriate teaching-learning environment is essential.

When special educators began using computers, they often started out with simple instructional games and drill and practice software (Mokros & Russell, 1986). Because using the computer was a new experience, and because the software often included colorful graphics and humorous sound effects, this software proved to be a powerful motivator for unmotivated students. At the same time, games and drill software were easy to implement for a busy teacher. Students could usually manage the software with little help, and teachers could devote their attention to other instructional tasks while the software took over.

As the role of computers in education expanded, new software tools moved into the schools, and educators developed a more sophisticated vision of the educational interaction between teacher, child, and computer. While drill and practice continues to be the predominant kind of computer instruction used with special-needs students, important new forms of software began to find their way into special education: word processing, problem-solving software, *Logo*, data bases, spreadsheets, and mathematics and science simulations. The emergence of what we refer to as "learner-centered software" has opened many doors for special education students and teachers. At the same time, it has made the selection and effective use of software a much more challenging task. It is the purpose of this handbook to make this task more manageable, and in doing so, to give teachers and students many more opportunities for meaningful learning.

WHAT IS LEARNER-CENTERED SOFTWARE?

When we began this book, we decided to focus on non-drill uses of the computer. While as educators and former teachers we are certainly not opposed to judicious use of drill, we saw in both our own work and in the work of other teachers that drill software tapped only a tiny part of what the computer can do. Some special education teachers whose first use of the computer was for drill quickly became dissatisfied. They sensed that the computer was more than a tool to do the same old things in a different medium, that it offered entirely new possibilities for their students. As one teacher put it, "I'm not even sure what it can do, but I have the feeling that there's more here."

Indeed, there is more here. When we look around at *society's* use of the computer, rather than at school use, we see what the significant applications of the computer include:

- storage, retrieval, and organization of massive amounts of information; subtle and complex computer graphics and animation
- programming languages through which new uses of the computer are created, from analyzing the authorship of historical documents to predicting the time and location of an earthquake

- simulations that model systems too small, too far away, or too inaccessible for study; telecommunications networks that link computer users and information.

The computer workhorses of the real world are not drills, games, or tutorials, but word processors, databases, spreadsheets, graphing packages, telecommunications, and other such flexible tools.

While not all of these real-world applications are yet easily accessible to teachers and students in schools, many of them do have educational counterparts. For this handbook, we wanted to compile information about significant computer applications that are being used successfully with special education students. However, defining these computer uses negatively as "non-drill-and-practice" seemed unsatisfactory, so we looked for a term that would describe this software. As we talked to teachers who were developing effective uses of non-drill software with their students, we found a consistent theme in their reasons for using this software. These teachers were all looking for ways in which their students could improve poor self-images, become more independent in learning tasks, take more responsibility for their own learning, and learn to view themselves as able to cope with the learning process, including its inevitable times of confusion, frustration, and difficulty. While these teachers also had specific instructional goals in mind, their search for materials that would teach *content* was inextricably tied to their desire for contexts in which students would learn to improve their learning *strategies*. After all, most of these teachers had been teaching for many years. They had already obtained or developed many materials and techniques for teaching content. What appealed to them about the interactive nature of the computer was the possibility for their students to become more independent learners.

Given the strength and consistency of this theme in what we heard from teachers—the concern that students become *learners*—we designated the type of software these teachers were choosing for their students as *learner-centered software*. Learner-centered software can include interactive games and tutorials, simulations, problem-solving software (including programming languages such as *Logo*), and tool programs such as word processors, databases, and graphics packages.

FEATURES OF LEARNER-CENTERED SOFTWARE

Learner-centered software puts *students* in control of learning activities and challenges them to generate and solve problems as diverse as writing a story, solving a mystery, or creating an animated picture on the computer screen. The child becomes actively involved in charting the learning course that he or she will follow while engaged with the software. Learner-centered software encourages the child to think and to use information to solve problems—or to create new problems to solve.

Learner-centered software is characterized by four important features:

1. **Learner-centered software offers students choices in selecting the goal of an activity, the strategies to reach the goal, or both.** Much educational software typically used with special-needs students is designed so that students must use a particular strategy to arrive at a predetermined goal. For example, spelling programs may encourage students to use only the strategy of memorization to attain a goal such as "90% accuracy in spelling words of X level of difficulty." In contrast, learner-centered software helps students to set their own goals. When using a word processor, for example, students decide what topic they will write about, who the audience will be, what information they want to convey, and how the paper will be structured.

   Other learner-centered software provides a predetermined goal for students, but encourages them to design and use their own strategies to reach the goal. For instance, in *Snooper Troops* (see Chapter 4), as students play the role of a detective attempting to solve a mystery, they are in control of decisions about locating information, selecting the information that is relevant, and using the information in one of a number of ways to solve the mystery.

2. **Learner-centered software provides feedback that is informational rather than judgmental. Students can use this feedback to expand their understanding of the content area.** Learner-centered software is based on the principle that students learn far more from *knowledge about the results* of their performance on a task than they do from simple reinforcement. Rather than providing simple signals of success and failure such as happy/sad faces or points on a basketball scoreboard, learner-centered software provides *usable feedback* which can help the student make a new choice. As educational researcher John Lochhead has observed, ". . . it is only by making (and recognizing) errors that real conceptual learning is possible" (1985, p. 6). Many special-needs students will give up in the face of negative feedback; many of those same students will keep persevering in the face of errors if they feel they can get closer and closer to a correct answer. Software that provides neutral, supportive, informational feedback is vital to their learning.

Consider the following drill and practice number-line game: A student must give directions to the computer to move a marker to a point on the number line by estimating the distance to the target point. The student gets only six chances to estimate, and once these chances are exhausted a "no more turns" message appears on the screen and a new target is presented. The student who does not reach the goal in six turns learns only that she has failed but does *not* learn how to improve her performance.

In contrast, many pieces of software do provide neutral, *usable* feedback. For example, a mathematics program such as *Power Drill* (see Chapter 2), which tells you whether your answer is too high or too low, helps you to shape your next guess. A reading comprehension program that provides the information that "The story says, 'It snowed all night, and there was no school the next day.'" helps a student to reread a selection to see where the correct information is found (*Gapper Reading Laboratory*; see Chapter 2).

3. **Learner-centered software allows, emphasizes, or encourages prediction and successive approximation.** In addition to providing usable information, learner-centered software is structured to encourage students to move toward closer and closer approximations of success. The software design is based on the assumption that students will make errors—and correct these errors—as they learn a new task.

The importance of learning through successive approximations is obvious in contexts such as sports. In learning to play tennis, the novice must *expect* to play clumsily and to chase more balls than he or she hits. But through repeated practice and observation of the effects of different swings, the novice player slowly begins approximating the goal of controlling the ball.

Learning to use fractions, to collect information for a report, or to write a coherent paragraph is not, in this respect, different from learning to play tennis. If the learner uses software to help master these tasks, the software should foster the development of closer and closer approximations of success. If, instead, the software penalizes "misses" while the student is involved in a learning process where errors are inevitable, the student may simply give up.

Underlying the principle of "successive approximation" is the notion that the student has an idea of what constitutes success: A student who is using a[illegible]tronic spreadsheet to plan the financial future of her bicyc[illegible]e is provided with a powerful tool for obtaining successively more realistic budget projections. The software tool encourages her to make multiple predictions, not by reinforcing her for getting correct answers, but by providing the instantaneous calculations she needs to compare, predict, and make decisions about how to operate a profitable business.

4. **Learner-centered software encourages learning within a meaningful context for students, building on students' intrinsic motivation.** Students learn best when they have a real reason for learning and when the goal of learning is relevant to their lives. Educators know, and research confirms, that when skill learning is not connected to a meaning-filled context, it is very difficult for students to learn these skills (Baroody, 1987; Ginsburg, 1977; Graves, 1983; Smith, 1985). A predominantly skills-based instructional program may contribute to students' passivity and lack of intellectual risk taking. Certainly, many special-needs students show this pattern—they can read the words, but do not know the meaning of what they have read; they can compute but cannot solve problems in which those computations are needed to find answers.

   How can learner-centered software make the learning environment more meaningful? First, as will be seen in this handbook, much learner-centered software provides rich and compelling problems to work on, problems like figuring out the strategy a spy is using to escape from you, or understanding how weather patterns determine the course of your hot-air balloon. This kind of software can take the student into a fictional world that is as vivid as a good novel, yet demands from the student good reasoning and an innovative application of skills.

   Other learner-centered software tools, such as word processing software or spreadsheets, provide a wide-open environment for students to work with their own ideas. With tools, the fact that there is no predetermined content to the software means that students are free to create a product that has personal relevance. For example, teachers find that students respond with interest and enthusiasm to *The Print Shop* (see Chapter 2), a program that allows students to apply writing skills by creating posters, greeting cards, banners, and stationery. In this context, children check their spelling, ask each other for advice, and design and try out their ideas in their own ways. Students are motivated to practice those skills which they most need because they are eager to make use of this opportunity to produce high quality work.

   While there may be explicit rewards built into some learner-centered software, this kind of software leads students toward internal motivation. Satisfaction is gained from a sense of mastery or completion. The finished product, the completed puzzle, the solved problem, or the new creation are the rewards from which students learn that they are able learners.

WHY SHOULD TEACHERS USE LEARNER-CENTERED SOFTWARE?

The first reason for using learner-centered software is this: *We need to create opportunities to be surprised by what our special education students can do,* and they need opportunities to surprise themselves. When we see them always—and they always see themselves—working in the contexts that are hardest for them, it is no wonder that negative self-images grow and our expectations shrink. When we see students in new contexts, we sometimes see a whole different person. One of the authors, while a classroom teacher, took her students swimming one June. Away from the schoolroom, Aaron, the worst reader in the class, uncertain and awkward with his peers, became a leader and a teacher. Who had known he was a champion swimmer outside of school? Inside the classroom, computer environments can provide contexts in which students surprise us. Stories throughout this book will show how such surprises can happen.

The second reason is related to the first: *We need to provide ways in which students can develop and use widely varying learning styles and approaches.* Many students learn differently from the way we expect them to learn—differently, perhaps, from the way we ourselves learn. Mary Briggs, a resource room teacher, recounts her experience with Norma, a 9-year-old learning disabled student. Mary had been making mobiles with several of her students as part of a special project. As the other students busily got started, Norma sat alone stringing a few beads together, apparently aimless and unfocused. "I kept urging Norma to get started until I realized *she was building the mobile in her head.*" From this and other similar experiences, Mary learned that Norma is a child who must see the whole before she can work on the parts, when she appears aimless and unmotivated, it is often because she has not yet grasped the problem as a whole. Learner-centered software provides ways for students like Norma to pursue similar goals, while using their own learning styles.

Third, *learner-centered software offers new options for teaching the critical processes of reading, writing, and mathematical reasoning.* There is software that emphasizes skills in organizing information, map reading, estimation, reading comprehension, and many other curriculum-related skills. The interactive nature of the computer allows students to try something, get appropriate feedback, and try again as they gain confidence and experience.

Fourth, *learner-centered software allows students to develop and use problem-solving strategies and thinking skills.* These skills, while acknowledged by all educators as valuable, seem slippery and difficult to measure. They are neglected in much of the software that is targeted for the special education market. Perhaps because *critical* *thinking* and *problem solving* are listed as "higher-level" skills on learning taxonomies, software stressing these skills is reserved for children who already have mastered a specified body of basic facts and figures. Unfortunately, we find that the considerable amount of learner-centered software that involves problem-solving and thinking skills is often aimed at the "gifted" student. We argue, along with many special educators, that students with special needs should be exposed to a wide range of problem-solving and learning strategies, both to help them develop new approaches to compensate for learning weaknesses and to encourage them to recognize and use their existing learning strengths. Learner-centered software can be a powerful and highly motivating vehicle for helping students develop problem-solving skills while at the same time leading to greater mastery of various subject areas.

Finally, moving beyond drill and practice *allows the special education student to begin to learn what the computer really has to offer*. It is a matter of educational equity and opportunity that special education students not be relegated to uses of the computer that involve only passive response, but that they learn to become active computer users. In the long run, for students who have difficulty memorizing or retrieving information, who cannot make their work clear and legible, who need environments in which they can easily change and improve their work, the computer is an essential tool.

Fortunately, special educators have access to a variety of learner-centered software that will enable their students to show individual learning strengths not tapped by the more traditional special education curriculum. To find this software, we must look beyond that which is targeted as "special-needs" software. To use it effectively, we must become familiar with applications for students with varying profiles of learning problems and strengths. The aim of this handbook is to help teachers who work with special-needs students find and use educational software that will truly enhance the learning process.

A GUIDE TO THIS HANDBOOK: WHO, WHAT, AND WHERE

This handbook is intended primarily for those who work with students who have learning problems. This group includes teachers who work in regular classrooms, resource rooms, and separate classrooms. We did not attempt to cover the communication and access issues of hearing-impaired, visually impaired, or physically handicapped individuals, but have chosen to highlight the cognitive issues that are relevant to many learners, whether they are classified as "special" or are simply having a tough time in school. We focus rgely on computer use in grades K-8, but many of the software applications, and many of the pedagogical issues involved in using learner-centered software, are equally relevant to high school teachers. Teachers in all these grades made contributions to this book.

This book is based on the experiences of many special educators who have explored the potential of learner-centered software with their students. Most of these teachers began with little computer experience. Through trial and error, careful observations of students using a variety of software, and much reflection and sharing, they discovered a great deal about what and how students learn when they use powerful software.

Teachers have made substantial contributions to this book by providing insightful accounts of ways that they have used learner-centered software, how their students have learned, and how their approach to teaching has changed as a result of using this software. We hope that what we have learned from them will suggest some paths for you to follow with your students.

We have structured this handbook so that the reader can easily explore different types of learner-centered software. If you are unfamiliar with this software, you may want to begin with Chapter 2, which provides a sampling of software that goes beyond drill and practice and at the same time fits easily into existing subject areas. Chapters 3 and 4 provide in-depth discussions of word processing and problem-solving software, respectively, with numerous illustrations of effective ways that teachers have used this software. In Chapter 5, we tackle the critical topic of using software to help students learn how to be better learners. As one teacher told us, if a teacher wishes to have any impact on a student beyond the time the student is in her classroom, the crucial task is to teach the student *how* to learn. Chapters 6 and 7 deal with the interaction between student, learning environment, and software—all of which is choreographed by the teacher. Besides serving as choreographer, the teacher plays multiple additional roles, as can be seen in Chapter 6. It is no accident that the final section of Chapter 7 deals with the nitty-gritty of getting started and finding support and resources. All software identified in this book is listed alphabetically in Chapter 7 with publisher, address, and computer type. We hope that by the time you finish the book, you and your students are eager to begin exploring learner-centered software.

NOTES

References for research and specific topics discussed are given at the end of each chapter. Teachers who want an overall, practical approach to computer use with special-needs children may find the book *Special Magic*, by Mary Male, of interest. For a more theoretical treatment, try Sylvia Weir's book, *Cultivating Minds*. Readers wanting to know more about the state of technology in education and its implications for instruction may want to obtain *Power On!*, a report prepared for Congress by the Office of Technology Assessment.

REFERENCES

Baroody, A. J. (1987). *Children's mathematical thinking: A developmental framework for preschool, primary, and special education teachers*. New York: Teachers College Press.

Ginsburg, H. (1977). *Children's arithmetic: How they learn it and how you teach it*. Austin, TX: PRO-ED.

Graves, D. H. (1983). *Writing: Teacher and children at work*. Portsmouth, NH: Heinemann Educational Books.

Lochhead, J. (1985). New horizons in educational development. In E.W. Gordon (Ed.), *Review of research in education 12*. Washington, DC: American Educational Research Association.

Male, M. (1988). *Special magic: Computers, classroom strategies, and exceptional students*. Mountain View, CA: Mayfield.

Mokros, J. R., & Russell, S. J. (1986). Learner-centered software: A survey of microcomputer use with special-needs students. *Journal of Learning Disabilities, 19*(3), 185–190.

Russell, S. J. (1986, Spring). But what are they learning? The dilemma of using microcomputers in special education. *Learning Disability Quarterly, 9*, 100–104.

Smith, F. (1985). *Reading without nonsense* (2nd ed.). New York: Teachers College Press.

U. S. Congress, Office of Technology Assessment. (1988, September). *Power on! New tools for teaching and learning* (OTA-SET 379). Washington, DC: U.S. Government Printing Office.

Weir, S. (1987). *Cultivating minds: A logo casebook*. New York: Harper and Row.

2. Familiar Content, New Approaches: Using the Computer to Support Skill Development

When computers showed up in classrooms and labs—often accompanied by little software and no training—many teachers were excited by the promise of the new technology. But how to use the computer well was not and still is not self-evident. It can take many frustrating hours to find the right piece of software for a particular student, to decide how to introduce and use it, to develop worksheets to accompany it, and to integrate the software with other parts of the curriculum. This isn't easy! In this chapter, we will share some ideas which have worked well as first steps beyond drill and practice. The learner-centered software in this chapter is well-suited to such first steps: It is clearly tied to familiar curriculum areas; it does not require extended time commitments for teacher or student; it is relatively easy to learn; and it requires less teacher planning and intervention than the software we describe in Chapters 3 and 4.

THE STORY OF ONE TEACHER

Many teachers who wanted to use computers could not find ways to do interesting work with the new technology. Although the computer made some things easier, special-needs teachers were provided, for the most part, with drill and practice software which they used to reinforce skills their students needed. Many found themselves comfortable using the computer in this way, but believed more could be done with such a powerful machine.

Cathy, a resource room teacher, complained, "If I have to help someone learn *Dieting Dinosaur* [a drill and practice program for practice in subtraction] one more time, I'll lose my mind! There must be more that they can do." Cathy sensed that her students were somehow missing an opportunity for appropriate uses of the new technology. She and her students had become tired of relatively mindless drill and practice work, and she needed something new.

Cathy explored other approaches to the computer that she felt were more compatible with her curriculum. She decided to concentrate her efforts on four fifth graders who came to her for help in math. She wanted to work intensively to teach them one or two skills which she considered important and which had presented problems in previous years. Cathy was looking for software she could learn easily and that, once integrated into her curriculum, would allow students to learn "something worth learning" on their own.

Coordinate graphing had been a problem area for these students. Using the mathematics textbook had never worked well for them, and Cathy had developed her own worksheets to help the students learn. The worksheets involved locating particular points, unscrambling coded messages, and finding points that could be connected to form a picture. Much to her delight, a computer program, *Bumble Games*, and its companion piece, *Bumble Plot*, were available for her to try in her resource room.

Having only one computer, Cathy assigned two students to use it while she worked with her other two pupils on the games and worksheets she had used in the past. Each pair of students used the computer every other day. Cathy initially had been concerned with one student in particular. Jeff hated the computer because, he said, it went too fast. His experience with arcade-style drill games had angered him. With *Bumble Plot*, however, he took command of the pace, playing hard, pounding the computer vigorously, and talking to it as he worked.

Interest remained high for a 2-week period, and Cathy found that the students were more motivated than in previous years. The skill levels of the students were improving and they seemed comfortable with coordinate graphing. Best of all, Jeff, who had not been well accepted in his regular classroom, became a successful and popular player of the coordinate-graphing board game, *Battleship*, in his homeroom. The fact that he had learned to play well using a computer added to the other children's admiration of his skill. The feedback Cathy got from Jeff's homeroom teacher was Cathy's first indication of a direct carryover into the classroom from the computer.

Cathy had been concerned about Jeff's unwillingness to take responsibility for his own learning. The passivity of special-needs learners can be quite dramatic, because they are reluctant to take intellectual risks, to try a new method, or to do something just to see how it works. After a school career studded with red marks on worksheets and filled with remediation, this is understandable. Consequences for students' learning behavior and self-image are severe, and every special-needs teacher wrestles with issues of negative self-image.

Resource room teachers know how debilitating this passivity can be. Not only does the learner have trouble in finding approaches to solve problems, but the unfamiliar presentation of material is likely to make the special-needs learner retreat into silence or off-task behavior. Learning becomes, therefore, not a problem to be solved, but a problem to be solved *in the way the teacher tells you.* Finding the right answer becomes more important than understanding: for many special-needs students, understanding schoolwork is exactly what they have *not* been able to do.

Appropriate use of the computer may allow the student to take control of his or her learning in new ways. A program such as *Bumble Plot* allows students to explore a topic at several levels of sophistication. A student like Jeff can decide the level he wishes to try, and can get feedback about success the moment he begins to do the work. Jeff knew that he was working well and was coming to understand what he was doing; Cathy respected that enough not to bother him.

Cathy was excited to find a piece of software that fit her curriculum and allowed a great deal of leeway for students' experimentation. She began to seek out more non-drill software which was easy to learn, appropriate to her students' learning needs, and fitted to her curriculum objectives. Cathy used the computer to provide a meaningful and enjoyable context for skill development and to put her special-needs students in the academic driver's seat. *Providing a meaningful context for learning* and *encouraging cognitive control by the learner* are two elements which can be strengthened by sensitive use of the computer and appropriate software with an involved and caring teacher. The software programs described in the rest of the chapter have this potential if used well. The next two sections highlight software in language arts and mathematics which teachers have found particularly useful.

THE COMPUTER'S ROLE IN READING AND LANGUAGE ARTS

We have often heard teachers express skepticism—and we have been skeptical ourselves—about what the computer can provide for teaching reading. "I can understand how the computer can help teach arithmetic, and I know that a word processor might help with writing," teachers beginning to use computers say, "but what can a computer do for *reading*?" We certainly do not want to use a computer to displace good literature as a primary means for engaging students in reading meaningful text. However, software has been developed which some teachers have found appropriate in supporting students' efforts to become fluent and competent in understanding written language.

Language arts software that appropriately supports language learning falls into three categories: (a) reading comprehension software; (b) language games which provide motivating contexts for practice; and (c) tools which provide structure and direction for writing. The next sections of the chapter provide examples of teachers' experiences with each of these types of software.

Reading Comprehension: Filling in the Gaps

Current research on learning to read and write reminds us that literacy is a total process. Smith (1975, 1985), among many others, stresses the interaction between the meaning of what is being read and the experiences the reader brings to the reading process. He highlights the need for making predictions and self-corrections continually as we read:

> Our predictions and hypotheses come from what we understand about the passage already; and our feedback, the information that tells us whether we were right or wrong, comes from what we go on to read. If we have made a mistake we will probably find out about it—and that is the way we will learn. (Smith, 1975, p. 98)

A cycle of guessing-testing-refining is essential to learning to understand written language. The software which we describe here supports that process. This reading comprehension software is based on the cloze procedure, a technique which has been used for extending reading competence. Cloze passages are reading passages in which selected words have been left out. The reader must pay attention both to the overall context and meaning of the passage as well as to the characteristics of the missing word in order to decide what word or words might make sense.

Mystery Sentences. A good introduction to software based on such successive approximation is *Mystery Sentences. Mystery Sentences* can present a sentence to the student in one of three formats: with every other word missing, with all consonants missing, or entirely blank. Students can see three clues to the sentence's content, ask for particular letters to be revealed, or guess letters or words, until they have figured out the whole sentence. Teachers and students can make their own files of sentences and clues.

Although the program comes with mystery sentences and clues which you can use, many teachers prefer to create sentences for students to guess. Teacher-developed mystery sentences can be based on student interests such as hobbies, musicians, or sports figures. Some teachers use social studies or science content, while others incorporate weekly spelling words. Current events interest many students. In one resource room, students worked on mystery sentences based on recent news events. Another resource room teacher used basal reading textbooks. Her students enjoyed trying to figure out sentences taken directly from a selection in their reading textbook. A third teacher emphasized language experience as students worked on sentences taken from *Charlotte's Web* by E. B. White, which she was reading aloud to her group.

Students can work with reading software in pairs. Dialogue between students working together encourages them to articulate their strategies and give good reasons for their guesses, as Dottie Spahr recounts:

> John was on the quiet side when I worked with him alone, but today he and Teddy carried on a lively dialogue about how to solve the sentences. They developed strategies together. They took turns typing and spelling; one typed as the other spelled the word. They used the context to pick a word and used the number of blank spaces to determine if it was a possible solution. As the game progressed, the boys became more animated. They started out sitting at the computer and ended up standing and making more contact with each other . . . . They helped each other with spelling and finding keys and most importantly in developing strategies. They were sorry they had to leave at the end of the half hour.

When students are comfortable with the program, they are often eager to develop mystery sentences of their own. Children write the sentence first and then the clues. Often writing the sentence and the clues on a separate piece of paper has helped the student remember the words when it comes time to put them into the computer. In one middle-grade resource room, students developed challenges for each other, and rules evolved as to how to make "good" clues. Their interest in writing grew as they developed exercises for each other.

What learning skills are developed in such a program? What benefit can this kind of activity have for a struggling reader? First, students are *using context clues* to develop their hunches. Second, they are *using phonetic analysis* as they use the "guess a letter" option. Third, they are *predicting* what the next letters or next words might be; those predictions can be tested by entering them as guesses. Fourth, they may be *using reference materials*. Students discover quickly that the use of the dictionary may help them; some students like to have their dictionaries ready so that they do not have to make guesses about spelling. Finally, in order to create clues for others to use, students must understand both the form and content of the material, thus *developing a solid grasp of the main idea of the sentence.*

Teachers who have used the program often move in a sequence from teacher control of the program to increasing student ownership. Because of its flexibility and modifiability, a wide range of uses is possible, and there are many ways to integrate *Mystery Sentences* into the curriculum. Donna Simone notes:

> I started with it as a group instructional tool—I found that to learn it alone is too hard. They start with one sentence and then can play against one another on one sentence. When they all can use it independently, I use their reading sentences or weekly vocabulary. Those at another level can create their own sentences and challenge others to do that. *The more that I use it, the more that I can find different ways of using it.*

The Gapper Reading Laboratory. While *Mystery Sentences* has been used successfully by teachers to engage reluctant students in reasoning about language, its text is limited to single sentences. In order to immerse students further in written text, we need *whole texts* (Holdaway, 1979) which provide a rich and detailed context for reading comprehension. *The Gapper Reading Laboratory* also makes use of the intrigue and challenge of "filling in the blanks" (or "gaps," in this case), but it provides complete reading passages and allows teachers to enter their own material or to use reading selections from the *Gapper Reading Anthology*. For each reading selection, the program offers a variety of activities: previewing questions, reading the entire selection (*Gapper* can keep track of reading speed), answering comprehension questions, replacing punctuation, cloze activities using the passage just read and, finally, the "Star Text." The Star Text section presents the entire selection again with only stars and spaces to indicate the letters in each word and the spaces between words. For example:

***** *** *** *** **** ******* ** *** ****.

would appear in place of the sentence, "Terry and her dog were walking in the snow." Students enter words that they know are in the story, and the computer reveals those words correctly guessed by the student until the entire passage has been uncovered. While this sounds daunting, remember that students have already read the passage at least twice and have answered questions about it. Once some remembered words are filled in, for example, Terry and *** dog **** walking ** the ****,

students can reason from the context—and draw on their memory of the selection—to fill in others.

The Gapper Reading Laboratory lends itself to myriad activities. Teachers report it to be a compelling motivator for many students in the middle grades. The game aspects present a challenge, and the self-competitive aspect of keeping track of one's own performance over time appeals to many students. Donna has students keep track of their scores on graphs over time. Most important, students are *reading* and focusing on the *meaning* of the text.

From the teacher's point of view, one of the critical features of *Gapper* is the ability to modify many aspects of the program. Teachers can control the speed of presentation, the words or types of words removed in the cloze procedures, and the amount of performance information presented to the student. Teachers can enter text and questions into the program, modifying the difficulty level of the material presented as well as the kind of feedback students receive.

Because both teachers and students can create their own selections and questions with *Gapper*, many extensions become possible. Either students or teachers can enter selections from books they enjoy, create comprehension questions, and decide on feedback for both correct and incorrect responses. The excitement of creating questions which other students will answer after reading the story led one student to say, "I never realized how hard it is to write a test. You have to know the whole story real well so that it's fair." Her reaction gives a sense of the motivating effect of being in control.

Teachers may enter selections from textbooks which special-needs students must understand. Charlotte Reid used *Gapper* to create review quizzes for her special-needs high school students. "When they came to me whining about their tests in their other subjects, I would sit down and create review sections from their textbooks. The last summary of the chapter was usually what I'd enter, and then base questions on it. They really loved it and we began to develop a library of review disks." Students can write stories specifically for inclusion in the class *Gapper* anthology. One class solicited manuscripts and chose their five favorites to include on an anthology disk.

Gapper and *Mystery Sentences* are examples of programs that support many uses while challenging students and presenting reading material in interesting and novel ways. Their flexibility supports students with a range of ages and reading levels. Both of these programs can support and extend off-computer work in reading comprehension.

Language Games: Playing with Words

Playing with words is an activity often engaged in by young children learning spoken language, but most of us continue to enjoy playing with words. The baby's babble, the 6-year-old's silly rhyming ("double, trouble, stubble, bubble, yubble, cubble, mubble, flubble"), the poet's limerick, the crossword-puzzle solution are all examples of play with words. Software which provides a context in which to play can support much incidental language learning. If you think about the word games you played on long car trips when you were growing up, you will have a good idea of what this software is like. Remember "Ghost," "I am Thinking of Something that Begins with -- B," "Hinky Pinky," "Categories"? While these games are fun, they also require some skilled understanding. As children listen to others' word choices, new words are learned and become a part of the game culture.

Much software that attempts to teach spelling or vocabulary directly presents isolated words for memorization with, at best, a single sentence as context. These activities often provide little motivation and few cues for learners. They require memorization and recall, preventing learners from using strategies to solve the problem. For a learning disabled student who has difficulty memorizing but knows a lot about the world, this kind of activity is a frustrating experience. Software described in this section provides a variety of playful contexts in which students can use and extend their knowledge of spelling and vocabulary.

Word-a-mation. Word-a-mation, for example, develops language skills by having students manipulate words in a word chain. Through successive transformations, the starting word is doctored by changing one letter at a time until it matches the target word. For example, can you change ROAD to LEAN with three intervening words? Here's one solution:

ROAD
LOAD
LOAF
LEAF
LEAN

Word-a-mation varies this idea by including other kinds of transformations, such as finding antonyms, synonyms, homophones, or words in the same category. A sequence from HAIR to COOL might be constructed as follows:

HAIR

> homophones

HARE

> change one letter

HARM

> change one letter

WARM

> antonym

COOL

Donna Simone uses this program with high school learning disabled students:

First, the students need some off-computer work that preteaches synonyms and antonyms. Then I start at the level where all five words are given by the computer, but they come up with the relationships. At first, they need cue cards next to the computer. Their big "aha" is seeing that—how can I put this?—if they take their hands off the keyboard and think *in their heads* they don't have to use hints provided by the computer. They can actually think up the answer themselves.

Hinky Pinky. *Hinky Pinky* is a language arts game which stresses rhyming and syllabication. The program has stored a number of Hinky Pinkies (word riddles) and has room for you to create your own. First, you are given a definition and a syllable clue, for example, "What's a Hink Pink for a little entryway?" The phrase "little entryway" is the definition and "Hink Pink" tells you that it is two rhyming words of one syllable each (a *Hinky Pinky* has two syllables in each word, a *Hinketty Pinketty* has three). You guess a "small hall," which is correct. Should you want to get clues from the computer you can get lists of rhyming words to choose from. This game is very appealing to many students. They learn to appreciate definitions and syllables, to extend their vocabulary, and to attend to details. Because they can enter their own Hinky Pinkies, they can be encouraged to practice rhyming and syllabication in a context of creativity and enjoyment, and a class file of Hinky Pinkies can be developed and printed out.

Word Quest: *Word Quest* is a very simple game to learn and can be used at any grade level if appropriately controlled. The game itself consists of two options. Option one is to create a word chain which goes on for as long as possible by entering a word that lies alphabetically between two others. If you start with *dog* and *guppy*, you might enter *flag*; then you will be challenged to find a word between *flag* and *g ppy*, and so forth. Sustaining the word chain becomes challenging as the alphabetical gap narrows. As students' vocabularies grow, their chains increase. Option two in *Word Quest* is to find a target word by narrowing down the options alphabetically. One player selects the word and the other guesses, coming closer and closer until the word is discovered.

Students use *Word Quest* to find specific words from the spelling list of the week; they use science or social studies vocabulary words to create chains. Because the chains can be printed out, the teacher can keep some record of the guessing process and can be alert to spelling errors. Young students can guess with letters alone, either creating chains with letters (What lies between P and Z?) or finding a mystery letter by narrowing in on it alphabetically. Older students can be challenged to create longer and longer chains which must contain no spelling errors.

What is *Word Quest* about? Primarily, it engages students in a particular search strategy—the same strategy we use for finding words in the dictionary, names in the telephone book, articles in an encyclopedia, or book titles in a card catalog. This gamelike practice with alphabetical order is helpful to many students. The structure of the activity offers good practice for the use of guide words. As the interval between words narrows, students encounter the more difficult alphabetizing rules. What goes between *slope* and *sloppy*? (*Sloping* is one possibility.) What goes between *slope* and *sloping*? (How about *sloped*?) In *Word Quest*, students frequently encounter suffixes, so rules for adding word endings can be stressed. This program presents a good context in which to practice all of these skills while students are challenged to learn and use new vocabulary.

Constructing Language: Tools for Creating Puzzles, Cards, and Stories

The third type of language arts software might be called *language construction tools*. These tools, unlike word processors, are not totally open-ended. Rather, they provide some direction and structure within which students can use language, and sometimes graphics as well, to create their own products. These tools provide a balance between creativity and teacher direction, which may be appropriate for a student who lacks confidence and fluency in the use of written language. These construction sets provide an easy way for students to begin to create with language as well as a final product which looks finished and professional.

Crossword Magic. For instance, *Crossword Magic* allows teachers or students to create their own crossword puzzles with their own words and definitions. Charlotte Reid initially used these puzzles to help her high school students master vocabulary for their science class. First, she designed puzzles which allowed students to fit given words into a puzzle, using definitions and spacing as clues. Next, she moved to using the weekly science vocabulary words without providing the list, using only the definition as a clue. Finally, the students made up their own puzzles, using their own definitions of the weekly words and clues. She found that the students' scores on their science tests improved dramatically as they used the puzzles for review. Not surprisingly, they were most successful in the weeks in which they made up their own puzzles.

Print Shop. *Print Shop* is a graphics program often used by teachers as a way of helping children take pride in their products. With a few keystrokes, students can create designs, border patterns, and brief texts for report covers as well as banners to announce classroom celebrations or commemorative cards for various holidays or events. Teachers have encouraged pen pals by generating letterhead paper with their students, who then, of course, want to write letters. The activities supported by this program require a good deal of student planning and proofreading and can develop students' interest in word processing. Other more complex programs (*Newsroom, Front Page*) can be used with older children to produce newspapers and other written documents, but *Print Shop* is a simple and exciting way to begin. The enthusiasm students have shown in their work with *Print Shop* has been the hook which has convinced many teachers of the usefulness of the computer for their students with learning problems.

Story Book Programs. Another combination of graphics and text is seen in the story book programs, with which students can create their own illustrated stories. These range in complexity from the relatively simple *Kidwriter*, with a limited set of words as well as pictures, through *Storymaker* and *Bank Street Story Book*, which combine graphics and simple word processing capabilities. Each of them provides some "canned" pictures and encourages students to draw their own pictures as well, using a joystick, mouse, or *Koala Pad*. Typically, graphics and text can be combined in any arrangement on the page. Such a flexible writing tool has an array of uses, ideally suited for student-teacher or student-student collaboration.

One of the most interesting software developments is D. C. Heath's *Explore-a-Story* series. What started as a small group of programs has turned into a much larger collection, spanning the grade levels 1-6.

One resource room teacher, for example, used the program *The Bald-Headed Chicken* with her fourth grade learning disabled students. This program consists of a story book to read to or be read by the students and a program which includes pictures from the story with automatically animated characters. The pull-down menu at the top of the screen allows students to alter the pictures by changing the scenery, and by adding or removing characters, objects, and labels for the pictures and objects. Students can also type directly onto the pictures they have created.

First, of course, the teacher read the story out loud to her class for enjoyment. Next she used the computer with the large video monitor in her room to show the illustrations on the software to the whole class at once. They all retold the story and the teacher demonstrated how to grab images of story characters using the mouse and move them into place in the pictures, or to remove them. When each scene was set up to their satisfaction, the students dictated parts of the story to her as she typed them onto the screen. When the class was through with this whole-group literacy project, they had created six scenes with commentary.

Next, each pair of students used the computer program to develop a sequel to the first adventure. This took a long time, since they worked on both the story line and the graphic details of the scenes. After each group had created their sequel, their work was printed out. Thus they ended up with five booklets which followed the original story. This kind of creative but structured software seems full of promise for the students most in need of self-expression yet still in need of support systems as they write.

Each of these tools is less flexible than a word processor, but their structure gives students who are reluctant to create with language a place to begin. It may be less terrifying for a child to write a greeting card using *Print Shop* than to begin a story on a blank word-processing screen. Beginning with a picture, using *Storymaker*, may be just the right path for a student who has a clear image of what she wants to say, but cannot find the words to say it.

Using such construction tools, students begin to enjoy the possibility of trying out their ideas, then changing and improving them, with no penalties and no laborious recopying. They find they are able to create an excellent final product rather than settling for one that is just barely acceptable. Most important, with support from a sensitive teacher, they can begin to bring into being creations which have originated from their own thinking and are designed to please themselves, not just to fulfill an assignment.

THE COMPUTER'S ROLE IN MATHEMATICS

Special-needs students present a challenge to a mathematics teacher. Often they are uncommonly skilled in some aspects of mathematics and disabled in others. Their strengths may show up in surprising areas, such as spatial visualization or problem solving. Specific problems in mathematics may be compounded because students are not taught mathematics in a variety of different ways and therefore do not have the opportunity to build mental images or models of the number systems which work for them.

The results of the National Assessment of Educational Progress (Dossey, Mullis, Lindquist, & Chambers, 1988) constantly remind us that while our students typically can do arithmetic computations, many do not know how to apply these computation skills in the context of solving problems. Most of our students are handicapped by an inability to use mathematics sensibly in real problem situations. Mathematics difficulties may be even more glaring for special-needs students. Because they are so far behind many of their peers, they have been encouraged to memorize rather than to understand what they are doing. Often students memorize arithmetic procedures, like borrowing or long division, but then do not select and apply these procedures appropriately to solve problems. Their informal knowledge of the way numbers work in the world is not tapped by paper-and-pencil work from the textbook. How can the computer contribute to their understanding?

Some programs available on the computer are excellent contexts for students to practice or develop their numerical problem-solving skills. Rather than presenting mathematics as a series of speed drills or procedures to be memorized, these programs emphasize the reasoning processes necessary to find a solution to a given kind of problem. Students can be in control of many aspects of these programs and typically can choose the difficulty or range of a problem. There are no time pressures, so that students have the opportunity to discuss problems cooperatively or to reflect on them individually without penalties associated with speed. Most important, the solutions can be found through a variety of methods, and the problems presented lend themselves to this variety. A teacher who wants to stress the value of alternative strategies can use the programs as a stimulus for students to compare their approaches.

The mathematics software which has been used with students having difficulty in mathematics falls into several categories which are analogous to the language arts categories used in the previous section: (a) mathematics comprehension software which stresses number sense and number relationships; (b) mathematics games which emphasize reasoning; and (c) application tools which provide contexts for the use of mathematical skills and the generation of individual mathematical ideas and theories.

Mathematics Comprehension: Building Number Sense

Elementary and middle school teachers often talk about one of the most important ideas in learning mathematics as "number sense." While this term may not appear in the list of skills in your textbook, it is one of the most crucial parts of early mathematics learning—the development of a sense of quantity. As one teacher remarked after observing a 7-year-old having serious difficulties in mathematics struggling with a simple problem: "I didn't feel he had a sense of what 24 was, because he couldn't work with it easily. It was like he had to really keep saying 24 to himself, and he couldn't break 24 up into 20 plus 4. . . .He couldn't just play with it. It was a number. It wasn't a *sense*; it wasn't the 'twentyfourness' of 24!"

In the primary years, students develop a sense of how big 5 is, then 10, then 20, then 100. Their developing sense of these quantities eventually leads to a sense of relationships among quantities: About how much of a hundred is 48? What would be a reasonable estimate for a sixth of 1,183? This sense of number should continue to develop through the elementary and middle grades as larger whole numbers, higher place values, fractions, and decimals are added to the mathematical repertoire. However, students with very poor number sense are all too familiar to us. They are students who may make errors such as the following:

35	1.5	1/2 + 2/3 = 3/5	35
+ 68	.5		x 43
913	6.5		105
			140
			245

They do not notice that their answers are not reasonable. Many of these students may have sound mathematical intuitions outside of school, but they do not believe their out-of-school knowledge has anything to do with the problems they face in school. The student who knows that four quarters make a dollar does not use that information when he or she must divide 100 by 4 in some other situation.

Estimation, judging the reasonableness of results, and having a sense of relationships among quantities are necessary abilities for all students. They are *critical* for the student who may never be good at memorizing facts and procedures. For some students, accuracy and detail can be ensured by the use of a calculator, as spelling can be helped by a dictionary or electronic spelling checker, but, even with a calculator at hand, students *must* grasp what numbers and procedures make sense in a particular problem situation. Some mathematics software emphasizes this "mathematics comprehension" by stressing estimation or by providing visual models for numerical relationships.

Fractions. *Fractions* (first developed as *Darts*) is one of the most studied pieces of mathematics software (Malone, 1980), and there is no question that it is effective in developing and supporting upper elementary and middle grade students' concepts of fractions. *Fractions* presents students with a vertical number line to which several balloons are fastened at unmarked points. Several integers—for instance, 1 and 2—are marked to give landmarks on the number line. Students' task is to name a point where they think one of the balloons is attached. If, for example, a student types in 1 2/3, a dart comes from the right side of the screen and hits the number line at 1 2/3. If the balloon is there, the balloon pops; if the dart misses the balloon, the point 1 2/3 is labeled on the number line so that the student can see exactly where his choice lies in relationship to the balloons. He now has another piece of information to help him make another guess.

This number line imagery is used in some textbooks, but is not much stressed. The dominant image in the textbooks is that of partitioning sets of objects or naming parts of one object. The number line imagery is important for many students because it stresses numerical relationships of fractions and mixed numbers in a way that is more comprehensible to some. Teachers have found that if they help the students walk through the first one or two problems themselves, they are able to use the program with little assistance because of the informational feedback provided by the software: Helen needed help in learning to find common denominators. She persisted in multiplying denominators and adding the numerators. When faced with a problem in adding fractions the teacher decided that he didn't have enough idea about how Helen thought about fractions, so he used *Fractions* as a diagnostic tool. He paired Helen and a good friend of hers and listened to the dialogue as they tried to identify points on the line.

From his careful listening, her teacher realized that Helen didn't know how to assign a value to a point. A point in between 3/4 and 1 was named 3/5 ("because 5 is bigger than 4"), and a point between 0 and 1/2 was called 1/1 ("because 1 is in between 0 and 2"). On the other hand, when she was given a piece of adding machine tape to fold into quarters and then into eighths, Helen very quickly figured that she could take short cuts. She enjoyed finding 5/8 of a long piece of tape and comparing it with 5/8 of a short piece of tape, and discovered that "the 5/8 size depends on the size of the piece it is 5/8 of!"

Suddenly this imagery system made sense to her and she used *Fractions* with great enjoyment. She became very accurate and fast and worked efficiently and well, first folding a small piece of string to approximate her answers and later using her hands to estimate. While her basic understanding of fractions was improving, her teacher reported that her addition of fractions also improved: "Somehow the denominators became more meaningful for Helen."

As Helen did, students sometimes develop their own measuring tools to use with the computer in order to make their guesses precise. A roll of adding machine tape or a piece of string used along the number line on the computer screen can be very helpful. The critical issue here is to support the student as she develops her own methods of "being sure," through trial and error, off-computer work, or the use of some on-screen measurement system.

Power Drill. *Power Drill* is another program which supports prediction and successive approximation in computation, in this case in the addition, subtraction, multiplication, and division of whole numbers. *Power Drill* presents students with problems of the form

$$24 \times ? = 5{,}640,$$

and students enter an estimate for the missing number. The computer shows them whether their guess was too high or too low by calculating the problem using their estimate, and placing that estimate above or below the original problem. For example, in this case, the student might guess 200, then, seeing that the estimate was too small, might make a new guess of 250. After these two guesses, the screen would appear like this:

$$24 \times 200 = 4{,}800$$
$$24 \times ? \quad = 5{,}640$$
$$24 \times 250 = 6{,}000$$

As students look for the missing addend, subtrahend, dividend, or multiplicand, they can use a wide variety of approaches—pure guessing, informed estimation, or paper and pencil strategies. Students, again, are in control of the parameters of the program, setting the operation and the range of numbers which will be used. For the older student especially, this piece of software provides an opportunity to try out answers which seem promising, and allows them to explore concepts of place value in a novel arithmetic environment. They can use pencil and paper or a calculator if they wish; the point of the work is to understand what is being asked for and to find some way to try to answer the problem. Even for adults, the thinking involved in solving the problem of 2,456 x ? = 31,928 is hardly trivial!

Joanne used this program with her middle school resource room students and she found that they liked to discuss their methods so that they could guess the answer in the least possible trials. However, their self-determined rule was that they were not allowed to use paper and pencil; all computation had to be done in their heads. They enjoyed using the results of their original guesses to refine subsequent guesses, and they became extremely good at making intelligent first guesses. "They developed what I would have had to teach them," she says, "except that they developed their own systems for making a good first try. When we do long division on paper, now, their first guesses are always quite close, and they have much less trouble estimating the relative size of an answer in any of the operations. Some of them still hate doing paper and pencil computations, but when they started to use the calculator, they had a good idea whether their answers were accurate or not; they can tell if it makes sense."

Students, like many of us, distrust estimation; they have been taught that estimation is a form of guessing and is therefore likely to be inaccurate. Worse, they may feel that the point of estimating is to be right on the first try! Perhaps the most important lesson they can learn from *Power Drill* is that estimation gives a great deal of information to the guesser, which can be folded in to continuing calculations and used as helpful feedback.

Each of the pieces of software mentioned here supports students' solving a problem in their own styles and with their own methods; each piece supports cooperative teamwork as well as competitive work; each involves practicing mathematical skills and ideas in a context that stresses number sense. Although students need to learn computational skills, those skills will not do them any good if they cannot decide how and when to use them, how to approach a problem, and whether an answer is reasonable. The ability to evaluate the results of a calculation is particularly important in this era of wrist calculators. Students will need to be able to know if their answer is even "in the ball park." Training in arithmetic must involve skills development in a broad context so that students are encouraged to try a variety of methods and to develop their own best approaches. In mathematics, as well as in reading and writing, *prediction* and *self-correction* are the critical skills to develop for future effective learning.

Mathematics Games: Playing with Numbers

Like the language games discussed earlier in this chapter, software that gives students a chance to play with numbers can offer opportunities to reason about mathematical relationships without requiring laborious calculations. In fact, one of the advantages of using the software described in this section is that the problems they present may involve small numbers and simple computations.

Students who fall apart when faced with a page of arithmetic or a set of word problems often choose numbers and procedures without giving much thought to the meaning of the problem. We are all familiar with the student who adds up all the numbers in a word problem, no matter what operation is actually needed, and ignores the unreasonableness of the result. When problems appear less threatening because the numbers are manageable, when computations are straightforward, when the student has access to a visual, interactive model of the problem and receives informational feedback at every stage, he or she is more likely to pay attention to the mathematical relationships in the problem and the effects of manipulations of these relationships.

For example, consider the following problem:

> You need to put exactly 5 gallons of water into a large unmarked pail in order to mix up a cleaning solution properly. You can easily get water from the bathtub faucet, but you have only one 4-gallon and one 3-gallon container. How can you measure exactly 5 gallons into the pail?

In the real world, estimation would probably serve our purposes well enough, but it is intriguing to try to figure out an exact solution to this problem (there is more than one). The numbers involved are small, and only very simple addition and subtraction are required. The problem is nevertheless quite challenging. (Some sample solutions appear at the end of this section.)

Puzzle Tanks. Puzzle Tanks is a piece of software based on this classic water-pouring problem. Students can choose problems at various levels of difficulty. They use a graphic model to actually manipulate the filling, pouring, and emptying of tanks as they try to solve a problem, so they can see their intermediate results as they move toward a solution. Many problems have multiple solutions. If students find a solution that works, but is long and tedious, they have the opportunity, if they wish, to try to find an alternative, more elegant solution for the same problem. Some problems cannot be solved, and students may choose to label them "impossible." The computer will tell them whether they are right that the problem cannot be solved or, if a solution is possible, will give them another chance to work on a solution.

Problems like this are ideal for cooperative problem solving. You can experiment with student pairs who work well together. Keep in mind that students who are "good in math" and students who are not can often work together in this kind of activity. You may identify students who have surprising strengths in this kind of reasoning, even though their textbook-based work in mathematics is poor as well as students who do very well in school mathematics but have poor reasoning skills. One fourth grade classroom teacher was shocked by the performance of her "best" math student: "She knows all her facts and she can do multi-digit multiplication and long division flawlessly, but she really had a hard time thinking something out for herself. All the stuff she's memorized didn't help her."

Quations. Other programs which emphasize numerical reasoning have similar qualities. *Quations* is a cross-number game, similar to crossword puzzle games, in which students receive a collection of "tiles." On each "tile" appears either a one-digit number or an operation sign (e.g., +, x, =). Students form equations on the computer screen, using their own tiles and connecting their equations to those that already appear on the board. While noncomputer forms of this game exist, this computer game offers several advantages. First, the students (up t can play) can choose to include a fictitious character, Qua' a player. While playing with Quato allows a single student to play with the computer, the presence of Quato as a player offers something even more important—models of the ways in which equations might be constructed. In effect, students can see a player who is expert, but not too expert, choose its moves. Second, the program evaluates student equations, so that students get immediate feedback about their responses. Third, you or your students can choose which operations are included (e.g., only lition and subtraction), so that the game can be made simpler , more challenging.

While *Quations* is appropriate for younger students as a context for using simple arithmetic facts, it also offers important mathematical content for upper elementary grade students, many of whom will be going on to junior high school with a very poor understanding of what an equation is. Many students infer incorrectly from their elementary grade experiences that the = sign in an equation means "now find the answer." For example, students interpret the equation 54 + 37 = ? in the following way: "The problem is, how much is 54 plus 37?; the equals sign means, write down the answer over here." Students with this understanding of equations continue to be stumped by problems such as 54 + ? = 91. Using their "now find the answer" interpretation of the equals sign, they will probably come up with 145. *Quations* is an excellent format in which to gain experience with the = sign as a sign of equality and balance. Since students are motivated to use as many of their number tiles as possible, in order to get more points, they learn that it is to their advantage to construct equations with more than one number on the right side of the equals sign, for example, 4 + 7 + 5 = 8 x 2. Including Quato as a player helps model this kind of equation.

The one disadvantage of the computerized version of this game is that students cannot physically rearrange their tiles in front of them as they plan the next move. Students can certainly try out their ideas on scrap paper, but paper and pencil may not be flexible enough for some students. Teachers have found that a set of numerals and operation signs on cardboard squares or index cards can be helpful; this simple aid allows students to make "rough drafts" of their equations before committing themselves on the computer.

Many other programs which emphasize reasoning are available. Look for programs in which the numbers and operations are simple but the problems are challenging; the feedback provides information and occurs at every step of the problem-solving process; and, if appropriate, graphic, interactive models show the results of each decision the student makes. What about transfer of such numerical reasoning skills to other situations? Research on problem solving (e.g., Zollman, 1987) shows that transfer of thinking skills usually occurs only for problems that are very close to the type of problems being worked on directly. However, direct transfer is hardly the point here. The purpose, rather, is twofold: for the student to find out about and learn to use the numerical reasoning skills he or she has, and for the teacher to see and support different facets of the student's intelligence. These insights can be used.

One girl was startled out of her usual whining during math class when her teacher confronted her with the expertise she showed in numerical reasoning: "You solved a difficult problem like the one you did yesterday afternoon on the computer, which hardly anyone else in the class had been able to do, and you're telling me you're not good in math? Let's see you use that same thinking ability that we both know you have for this problem. Come back in 5 minutes and you should be able either to tell me how you think it should be solved or explain to me exactly which part gets you confused."

(By the way, did you think about how to solve the 5-gallon problem? One possibility is to fill the 4-gallon container, pour as much as possible into the 3-gallon container, leaving 1 gallon in the 4-gallon container. Pour this 1 gallon into the pail. Repeat this procedure 5 times, and you will have measured 5 gallons into the pail. A more elegant, and shorter, solution is the following: Fill the

3-gallon container and pour the three gallons into the 4-gallon container. Fill the 3-gallon container again and pour into the 4-gallon container until it is filled. You will have poured one gallon off, leaving 2 gallons in the 3-gallon container. Pour the 2 gallons into the pail; then fill the 3-gallon container again and pour it into the pail, giving you a total of 5 gallons in the pail!)

Applications Tools: Constructing a Mathematical Environment

Mathematics tools, like language construction tools, allow students to put together original mathematical information and ideas. This software supports mathematical activities in which students formulate questions, make decisions, gather data, organize and display their information, and come to their own conclusions.

A number of programs allow students to create tables and graphs, including *Exploring Tables and Graphs* and *MECC Graph*. In one upper elementary classroom, the teacher divided the class into small groups of three or four students. Each group chose a survey question, canvassed the rest of the class, then used graphing software to display their data more clearly so that they could report on what they had found out. One group of students found out about their classmates' collections; another group gathered data on Friday night bedtimes; a third group investigated favorite academic subjects; and so forth.

One of the important aspects of this activity was that all students, including those with learning difficulties, participated equally in making decisions, using mathematical data, graphing their information, and drawing conclusions. While all students enjoyed using the software, the availability of this graphing tool was particularly important for students with poor organizational strategies and/or poor fine motor coordination. Drawing a graph by hand would have required all of their concentration and energy, might have led to needless frustration, and would have prevented them from paying attention to the important mathematical aspects of the problem. With the bar graphs they made with the software, they could *see* their results in a clear visual format and were able to spend more time reflecting on the meaning of the statistical information they had collected.

For example, the group which worked on Friday night bedtimes noticed that girls, in general, were going to bed much earlier than boys. The discussion which arose from this finding led not only to many theories about reasons but to a further survey investigation seeking more detailed information. These students were learning that they could derive a better understanding of real phenomena through collecting and analyzing mathematical information, and they estimated and used fractions and percentages in the process.

The graphing software available for classroom use is not yet as flexible and sophisticated as it is likely to become within the next few years. It does not approach the power and ease of use of adult graphing software in the way that, for instance, word processors available for classrooms approach the power and ease of use of adult word processors. Nor can you easily use adult graphing programs; few have been written for the Apple II to date. However, even what is currently available provides worthwhile tools for classroom mathematics and, if you have a printer available, a finished, professional-looking product reflecting your students' mathematical work.

Electronic spreadsheets are also becoming increasingly available for classroom use. Like an accountant's paper spreadsheet, electronic spreadsheets provide rows and columns in which text and numerical material can be organized, manipulated, and displayed. What the electronic spreadsheet adds to the old-fashioned paper-and-pencil method is many automatic functions, including arithmetic calculations. So, for example, if you are calculating your monthly budget, you can have the spreadsheet automatically compute percentages, subtotals, and totals. This allows you to play with your budget, to ask "what if?" questions. "What if I spend $25 more on entertainment and $40 more on clothing; will I still have enough to cover my fixed bills?" "What if I get a 5% raise; can I afford a more expensive apartment?" As you type in your new figures from the keyboard, totals and other calculations you have built into the spreadsheet are automatically changed so that you can immediately see the results of your speculations.

In one high school mathematics class for special-needs students, Michael Brandmeyer did just that, using the *AppleWorks* spreadsheet. He set up the electronic spreadsheet in advance with categories and calculations, then had his students use it to create a budget, given the possible earnings and expenses they might have in a few years. They researched jobs they might hold, given their probable level of education and training, and found out what they could expect to be paid. They scoured the newspaper to choose living spaces they could afford to rent and estimated a food budget by planning and pricing a weekly menu. As students learned about how to use the spreadsheet, Michael had to prestructure the spreadsheet less. Again, the spreadsheet provided assistance in organization and calculation, allowing the students to concentrate on *understanding* and *using* the mathematical information they collected.

Typically, spreadsheets are used with high school students, but they are also beginning to be used with elementary and middle school students. A self-contained class of 9- to 11-year-old special-needs students with learning problems used the same *AppleWorks* spreadsheet to record and organize information they collected. To the surprise and delight of their teacher, who was unsure whether they would be able to use this tool, they had little trouble learning how to enter information and how to use the arrow keys to move around on the spreadsheet. While the activities they did were much simpler than the high school budget project, they also benefited from a format which allowed them to record, order, and see numerical information clearly. This application is used by few teachers as yet, but holds great promise for the special-needs student. It offers a tool in mathematics comparable to the word processor for writing, and its potential has yet to be exploited to the same extent.

Another type of tool software is represented by *TimeLiner*. In this program students enter dates and events (1978, "I was born"; 1963, "President Kennedy was shot"; 1620, "Pilgrims land at Plymouth"), and the computer generates and prints out a formal, ordered timeline. This tool is used for a wide variety of purposes such as social studies, creative writing, and problem solving.

No discussion of mathematical construction tools would be complete without a mention of *Logo*. Although we have chosen not to cover *Logo* in depth, given its complexity and scope, those of us who have used it find it to be an excellent mathematical environment for the special-needs student. Temple Ary, a mathematics teacher at a school for dyslexic students, uses *Logo* in her mathematics classes to teach many arithmetic concepts, including multiplication, division, and equivalent fractions (Ary, 1987). Many other special-needs teachers use *Logo* as an environment in which students plan their own projects and, in the process, develop skills in number sense, estimation, angles, directionality, and sequencing (see Russell, 1983; Weir, 1987). Mary Briggs, a resource room teacher in an elementary school, recounts her experience with one of her learning disabled students. Using *Logo*, she had given her students procedures for three different sizes of triangles which they were combining to make patterns and pictures: "Kids noticed they could create the identical thing small as they did large, so *scale* emerged as an interesting item for them." One student, Brian, carried this idea of scale into his mainstream classroom, where they were working on mapping skills. Mary noted, "He totally made the connection about what scale was, even though we had never really talked about a map at all. The idea of the identical thing at a smaller size, how to do that, is something I think [he] really internalized."

In mathematics, students are too often required only to produce a single, right answer. They rarely have the opportunity to think about meaning, examine number relationships, play with numbers, or construct their own mathematical creations. The software in this section gives examples of activities and environments which encourage students to grapple with the meaning of the numbers they are using and to take more responsibility for making decisions using their mathematical knowledge. At the same time, it provides teachers the opportunity to see the variety of mathematical abilities which their students may have to offer.

THE TEACHER'S ROLE

The first steps are always the hardest, but the kind of software we have been describing lends itself to easy integration into the classroom. The skills being taught are straightforward and mesh with existing curricula very well. The advice of most special-needs teachers we know has been simply to begin—to find one good, flexible piece of software and to start. That may not be the best way to proceed with other kinds of software whose uses are more complicated (see Chapters 3 and 4), but the weight of advice is on the side of trial and error with skill development software. Here is the voice of one experienced special-needs teacher:

> You have to try it first with a child. You can't just preview it yourself. We tend to get more hung up in manipulating things than they do. They're more end-result oriented and they want to get through faster than we do.

The information which can be gained from observing students actually using the computer is invaluable and must form the basis for further curriculum planning. Embarrassingly often, our predictions about software or computer use will be wrong. We have to be willing to try new materials with specific students in order to plan well for their use. Many teachers worry about trying out software with students when they are not experts in computer use themselves. However, many teachers find that if they are willing to adopt an experimental attitude toward software and to try things openly with their students, the students begin to adopt a much more open and exploratory attitude toward their own learning. One teacher put it this way: "After I let them know that I was a beginner, they relaxed. It seemed to make their mistakes all right, too."

However, keep in mind that complexity generally increases as you move from software that presents games or structured comprehension problems—the first two categories of software presented in the language arts and mathematics sections of this chapter—to software that provides construction and application environments—those presented in the third category of the mathematics and language arts sections. This tool software offers more possibilities, variation, individualization, and opportunities for creativity and invention.

However, its larger number of options may initially be more difficult to understand. Each of these pieces of software usually has a set of conventions which students must get used to before they can use it well. For example, when using *Explore-a-Story* software, students need to learn how to pull labels and pictures out of storage and how to move objects on the screen. *Exploring Tables and Graphs* requires users to learn a somewhat awkward method of inserting information into a table.

Learning different conventions can be clumsy, as one special education teacher mourned: "Why don't you get the message to the software manufacturers to use the same conventions in all their software?" The fact is that interaction with software is becoming easier and easier, but many of the improvements have not yet made their way into the computers and software used in schools. However, it is also true that students, including students with special learning needs, often have much less trouble learning to use the software than we expect. It is they, not we, who have grown up in a technological age and it is they, not we, who are most ready to become involved with technology.

USING SKILL DEVELOPMENT SOFTWARE: WHAT DIFFERENCE DOES IT MAKE?

It becomes clear in talking with computer-using special educators that there are two areas of gain: *for the teacher* and *for the students.* Teachers often report that they have found interesting new options for presenting concepts and skills or reaching previously uninvolved students. These gains in curriculum planning may be a by-product of their new sense that they can see and effect change in some students' learning. Teachers who are willing to experiment and try new ideas in their teaching are rewarded by an increased sense of efficacy and by unexpected insights into their students' learning. Some teachers report that they feel renewed excitement about their teaching because they have begun to look at students in a new way. Rather than seeing them as a collection of remediable deficits, one notes that she is now planning for her students based on their interests and their strengths, and that her teaching is much more enjoyable than in the past. "I've found some software programs which match with the kids' interests and my teaching as well as with the curriculum we have," she says, "and it's so logical that I can just go about my business. No one challenges me."

Students, on the whole, respond positively to software which presents them with reasonable challenge and which is presented as a serious and interesting learning opportunity by the teacher. At all grade levels, special-needs teachers who use computer software are surprised by some students' responses to it. Some students learn more and more quickly than expected, and some just "don't get it." However, as teachers become skilled at selecting software appropriate for students' different learning styles, they report more and more instances of students who are "turned on" to learning for the first time, or for the first time in years.

Good skill development programs are never going to take the place of teacher instruction, but they can be a catalyst, providing support and motivation, offering new imagery systems and encouraging increased student control of learning. Here are comments from several teachers:

> I was very pleased with the cooperation that *Mystery Sentences* promotes. It was fun to see the pairs of boys help each other and in some ways compensate for each other.
>
> John was actively engaged for a very long time. He was pleased with his ability . . .
>
> Drew seemed more assured of herself this week and did not state her answers questioningly. She enjoyed working through the doze portion of the program and was able to work independently.
>
> *Print Shop* . . . was terrific. The children made invitations to our Thanksgiving pancake breakfast and they all made Thanksgiving cards for their families. We also made banners and some letterhead stationery. It's a very easy program and the children and I absolutely love it.

Extravagant claims have been made for computer use: It will motivate the unmotivated, teach the unteachable, help those who resist help. These must be taken with a grain of salt, but teachers consistently report strong evidence that there are gains in many areas from computer use. Teachers often feel better about their teaching and themselves. Students work hard and sustain attention on projects from day to day. Work is shared and students cooperate with each other. Students keep records of their progress in which they take pride. Easily distractible students show attentional gains.

The effect of using skill development computer software on any individual student, however, is measured best by that student's performance and attitude over the long run. Matching a student's strengths and interests with the proper activity (computerized or not) in an interesting context can produce real cognitive gains. The computer allows us a context within which interest can be created and sustained, if proper software is found and matched appropriately to student needs.

NOTES

In addition to the skill development software described in this chapter, a wide array of other programs is available. Titles include *Missing Links, Wizard of Words, Super Print, Number Quest,* and *How the West was One + Three x Four.* All are listed in Chapter 7 with publisher, address, and type of computer supported.

REFERENCES

Ary, T. (1987, April). The giant inch and other Logo programs for math class. *Classroom Computer Learning, 7,* 57–59.

Dossey, J. A., Mullis, I. V. S., Lindquist, M. M., & Chambers, D. L. (1988). *The mathematics report card: Are we measuring up?* Princeton, NJ: Educational Testing Service.

Holdaway, D. (1979) *Foundations of literacy.* Portsmouth, NH: Heinemann Educational Books.

Malone, T. W. (1980) *What makes things fun to learn?: A study of intrinsically motivating computer games.* Palo Alto, CA: Palo Alto Center. Xerox.

Russell, S. J. (1983). Had we but world enough and time: Logo in special education. *Classroom Computer Learning,* 4(3), 34–39.

Smith, F. (1975). *Comprehension and learning: A conceptual framework for teachers.* New York: Holt, Rinehart & Winston.

Smith, F. (1985). *Reading without nonsense.* New York: Teachers College Press.

Weir, S. (1987). *Cultivating minds: A Logo casebook.* New York: Harper & Row.

Zollman, A. (1987, April 20–24). *Aspects of transfer of learning in mathematical problem solving with respect to the order of problem presentation.* Paper presented at the annual meeting of the American Educational Research Association, Washington, DC.

3. Using the Computer to Teach Writing

Many special-needs students have problems with writing—problems which surface at different levels. At the mechanical level, students have trouble with spelling, punctuation, and handwriting. They may be unable to construct grammatically correct sentences. On another level, perhaps because of their problems with the mechanics of writing, special-needs students have difficulty focusing on the flow of ideas in their writing. The stories they write are typically shorter and more barren of detail than the stories written by their classmates. Because of their lack of writing skills, many of these students appear disinterested in writing, and lack self-confidence in their writing ability.

In the words of teacher Marguerite Nelson Creskey, for students with these problems "the word processor is the equivalent of a ramp for physically handicapped individuals." It helps students get around their writing problems and gives them access to written expression. Teachers have found that the word processor helps students focus on their strengths as writers and allows them to find ways of communicating more effectively. Like other children, special-needs students have a variety of interests and ideas, as well as the desire to communicate about them. The word processor allows children to demonstrate their expertise by writing fluently about meaningful topics without stumbling over mechanical obstacles. At the same time, the word processor allows teachers to spend time thinking about how to facilitate writing, rather than simply serving as proofreaders.

While it is no panacea, the word processor can help special-needs students gradually begin to tackle problems with writing. Ultimately, it can help students be more confident writers who enjoy using their writing skills to communicate with others. By giving them a vehicle for writing frequently and writing about things they care about, word processors help students learn not only writing skills, but also grammatical and spelling skills. Some teachers find that because students are able to read their own writing when it appears on the word processor, they are better able to focus on how the words should look and be spelled.

Research has demonstrated that children learn more about the mechanics of writing *by writing* than by doing workbook exercises on grammar, punctuation, or spelling (Graves, 1983; Graves & Calkins, 1980). Formal instruction in grammar has no lasting effect on the quality of students' writing. By giving students more opportunities to write and rewrite, we help them improve both their composition skills and their use of standard writing conventions.

Some educators think that the greatest potential of word processing is with students who have an aversion to writing or severe problems with the mechanics of writing. These students stand to gain the most from a tool that will simplify the writing and revision process. As one parent (herself a school principal) commented about her learning disabled seventh grader: "He had never written a paragraph before this year. Then, when a teacher showed him how to use the word processor, his writing just took off. He's writing stories now. He doesn't think he's dumb anymore."

PRACTICAL ADVANTAGES

In addition to its great potential to improve the writing and self-confidence of special-needs students, word-processing software has great practical advantages. One major advantage is that it meshes well with existing writing curricula. Teachers can easily transfer writing, editing, and grammar exercises to the computer, using the same instructional methods that they have used all along. It is easy to type, save, and print assignments on the computer, and the resulting product is pleasing to the eye of both student and teacher. Here is an instance where the transfer of existing learning activities to the computer is straightforward and effective.

A second practical advantage of word-processing software is that it can easily be integrated with the learning objectives that appear on students' Individualized Educational Plans (IEPs). When a teacher introduces word processing, few changes need to be incorporated into existing IEPs. The word processor is simply a better tool for accomplishing the writing goals that have already been established for individual students who have problems with writing.

In the classroom, it is difficult for teachers to encourage students to work systematically on planning, writing, and rewriting. In the time allotted to writing, most students have time only to create an initial draft, make minor corrections, and then copy onto white paper a cleaner version of the same draft. A major advantage of the word processor is that it buys more time: Once a piece has been entered, it can be reworked and embellished very efficiently. Because new work can be added without recopying the entire piece, the possibility of making new errors is largely eliminated. Freed from the burden of recopying papers to make them look good, students have more time to play with ideas, sequences, images, and language.

LEARNING OBJECTIVES

> [My goal] is to help the children reflect on themselves and their world and to help them express their thoughts and ideas in a coherent manner, where they're able to pay attention to some rules for punctuation, capitalization, and spelling. But I think it's more helping them understand who they are and be able to express that.

Betty Church, the teacher quoted above, has been using the word processor with her 5- to 10-year-old resource room students. Through activities such as writing poems about food, creating books about dinosaurs, and authoring autobiographies ("The Maze of My Life," as one student titled his essay), students are developing a sense that "the writing process is a very thoughtful, reflective, and personal process."

Word processors are most valuable when used to focus on what the child wants to say, rather than on the mechanics of writing or of using the word processor. Above all, the teacher must keep in mind that a central learning objective is to help students take ownership of their writing. Students who feel that they have something important to say to a particular audience are often more invested in saying it correctly. On the other hand, when the piece belongs to the teacher who has structured the assignment around a topic she has selected, students often do not care enough to put much thought into the writing assignment.

While it is tempting to plan several introductory sessions to master word processing commands, students are often bored by such sessions. Many teachers prefer to have students master the correct procedures as the need arises within the context of their own writing. In this context of working on meaningful writing, students often swap word-processing commands that they have mastered with each other. They learn to turn to peers for help when encountering trouble with deleting, moving, or other editing tasks. Teachers who encourage this exchange find that everyone benefits. On the other hand, teachers who insist that each word-processing command be mastered before proceeding with real writing find that students may lose interest before they ever get a chance to write.

Should editing and proofreading skills be stressed as a major learning objective? Certainly the word processor facilitates skill development in these areas. The problem is that for children who are not yet confident writers, editing gets in the way of composing. Children begin to focus on getting it "right," rather than on the process of writing. They begin making safe choices—for example, using the word "bird" instead of "downy woodpecker"—because they fear that potential spelling mistakes matter more than careful selection of words.

Children quickly become discouraged and easily lose their trains of thought when their thoughts are interrupted by a teacher's requests to go back and fix punctuation or spelling in a sentence that was just written. A major strength of the word processor—the ability to make instantaneous revisions—can become a drawback if teachers focus on corrections without attending to the content.

The issues are timing and emphasis. Word processors are best used for composing, without calling attention to the student's mistakes until drafting and refining have been finished. At a later stage, when students are proud of the content of their product, they are more than willing to polish a piece for publication. Keep in mind that simply in rereading their piece to see if it suits them, students will identify and fix many mechanical errors.

Students certainly can be encouraged to use spell-checker software to correct mistakes in spelling. Some word processors, such as *Bank Street Writer III*, include spell-checkers in the program; spell-checkers are also available separately (*Bank Street Speller*, *Sensible Speller*, for example). When they have finished writing, students use the spell-checker to identify words that have been spelled incorrectly. The software gives the student a list of words which it does not recognize, and the student takes responsibility for correcting the spelling mistakes. Note that many spell-checker programs include a dictionary to which students can add proper names and idiosyncratic words they use in their writing. Keep in mind that seeing and using the version of a word in a variety of contexts is the most effective way to learn its spelling. One student used the Find and Replace functions of the word processor to help him with his difficulty in spelling *Prometheus*. Every place where he needed to write *Prometheus*, he simply wrote *P* temporarily. Then he went back and instructed the word processing program to substitute *Prometheus* for each *P*. In this process and in rereading his work, he saw the correct spelling again and again. The more children write, the better their spelling skills will be.

SOME SIMPLE PREREQUISITES

What skills do students need to master before they begin using word processors? Although students need reasonably sound letter-recognition skills before they can start typing text into the computer, many students enjoy dictating stories to a teacher or aide, who types the story on the computer. Primary grade teachers and even some preschool teachers use the word processor to publish children's stories. Beginning writers and readers, even those who cannot yet recognize letters on the typewriter, get a great deal of satisfaction from the fact that someone values their thoughts enough to type and publish them. Reading and rereading these class books leads to the building of sight-word vocabulary as well as developing a feel for linguistic sequence and syntax.

When children use the word processor themselves, they are both writers and typists. This dual focus can be a problem for students who have special needs. What level of typing skills do children need in order to effectively use word processors? Educators vary in their answers to this question. Children as young as 6 can learn the positions of keys and how to use them in a hunt-and-peck fashion. In fact, finding and pressing a letter on the machine may be less difficult than "producing "the letter by hand, especially for children who have difficulty with fine motor skills. Typing ability clearly varies from child to child, and students who have difficulty with it should not feel like they have to become touch typists before they can even start writing. But it is clear that children who have mastered some basic keyboarding skills are more able to focus on their *writing*, with less interference from the *mechanics* of typing.

When children are left to their own devices at the keyboard, they typically create their own idiosyncratic methods for typing. One educator reports observations of 4- and 5-year-olds who had designed a quite ingenious system: They divided the keyboard into quadrants, moved their dominant hand to the appropriate quadrant, then searched for the appropriate letter within the quadrant. The only problem with this system is that children might develop habits which would later interfere with learning touch typing. Rather than leaving students on their own to figure out how to type, many teachers suggest that time be set aside for separate practice sessions devoted to familiarizing children with the keyboard. Keyboarding practice can be an ongoing part of computer education beginning in the early grades, but mastery of keyboarding skills does not need to be a prerequisite for using the word processor.

Sound keyboarding skills for beginners can be developed through the following methods which have been suggested and tried by teachers.

- Provide students with a cardboard template of a keyboard, which they can refer to and practice with during free time.

- At first, encourage students to keep both hands on the keyboard, using their right hand for keys on the right side of the keyboard, the left hand for keys on the left.
- Later, have students keep their index fingers on the "home keys," which are marked by a raised bump on many computers or can be marked with colored labels.
- Have students practice typing skills by using one of the keyboarding software programs for a few minutes before they begin writing. Be aware that students can become very adept at arcade-style typing games without using appropriate keyboard positions. Software that mirrors traditional, systematic techniques for teaching typing appears to be the most effective. Examples of popular keyboarding programs are *Type to Learn; MicroType: The Wonderful World of PAWS;* and *Success with Typing.*

FIRST STEPS

How do teachers begin to use word processors with their special-needs students? The first step, of course, is to become familiar with the software yourself. This means trying it out for your own writing, experimenting with its capabilities, and going through the minor frustrations that students are sure to encounter. Most teachers do not find it helpful to begin by reading the instructional manual from cover to cover. Use the manual as a reference book, rather than a step-by-step guide.

It is often a good idea to work through the tutorial software that accompanies many word-processing programs. Well-structured tutorials can give the new user an immediate feel for the software and what it can do. But even tutorials are no substitute for actually using the software to do some writing of your own—a letter, weekly plans, or notes about one of your students.

The best way to learn and remember the editing conventions is through a great deal of hands-on practice. Finding someone who can answer questions—another teacher, a friend, a student—has been critical for many teachers learning to use word processors. There is nothing more frustrating than getting completely stuck and feeling like you are totally inept at using computers. There is probably a simple way of getting "unstuck," and someone with a little more experience can often help you find your way out quickly. And it's much easier to ask a friend than to find the solution in the manual! Like learning the rules of a board game, learning a word processor's features is best done by "playing the game," and having others help you learn it.

After two or three practice sessions, most teachers feel comfortable enough with a word processor to introduce it to students. Thorough mastery of the software's features is *not* a prerequisite for student or teacher use. A few basic functions—how to enter text, how to delete text, how to insert text into what you have already written, how to save your writing to a disk, how to retrieve your writing from the disk—are needed to begin; additional mechanics, such as moving a paragraph or centering a heading, are best learned as the need arises. Many teachers find that *students* themselves become a valuable source of expertise, and are quite willing to share their expertise with others. In one classroom, two learning disabled junior-high students learned how to use *AppleWorks* on their own, although their peers were using the less complex *Bank Street Writer*. Their teacher commented, "It's wonderful. They're teaching me now—I don't even know how to access their files!"

TEACHER-TESTED WORD PROCESSING ACTIVITIES

Teachers suggest starting with relatively simple activities that involve the entire class. For example, a good initial activity is writing a group story. Each student takes a turn writing one line of a story. The finished story is then read aloud, and students critique it. Which parts lack clarity or detail? Which sentences don't "sound right"? This kind of discussion gets students used to thinking about writing as *communication to an audience*, with the focus first on sense, style, and vocabulary, and later on spelling and punctuation. Yet because the story is not the work of one author, the risk in hearing criticism is not so great. Once the group story is completed, it can be revised, expanded, or continued by individual students as a hands-on introduction to word processing.

Once students have had some practice with the word processor, they will want to work on their own pieces. It's important to give them access to the word processor for planning and initial drafts, as well as for revising. When the word processor is used simply as a typewriter, with students writing out the first draft by hand, the mechanical advantages of word processing are minimized. In all cases, the teaching goal should be to have children focus more on their writing and less on mechanical obstacles.

Below are some successful, teacher-tested writing activities that can be introduced to children who have used the word processor a few times:

- Have each student write a "book" to contribute to the class library. With younger students, use software that has extra-large print (such as *Magic Slate*) so that there will be many pages in the book. Students can make special covers, illustrate each page, laminate the pages, then share their contributions with others through the library. Children can check out these books to take home and share with parents. Unmotivated students who "don't have anything to write about" have often become interested when they begin to see the published ideas of their classmates.

- Write a class cookbook. Each student contributes a recipe, along with a paragraph about the dish (why it was selected, what's special about the dish, etc.). Once the recipes have been collected, the class needs to organize the cookbook and decide how recipes should be grouped. This is a good activity for helping students master word-processing features like setting margins and tabs, moving text, and formatting. It also provides a context in which students can develop good organizational and classification skills.
- Write journals. Give each student a disk and set aside some time for ungraded personal writing. Many teachers say that this is an extremely effective way of helping students with low self-esteem to begin thinking about themselves as writers. Journal writing has the additional benefit of encouraging students to articulate issues that are of personal concern.
- Use a word processor in conjunction with a graphics program to have children write illustrated stories or reports. For example, one teacher's class wrote and illustrated reports on silkworms using a combination of a word processor (*MacWrite*) and a graphics program (*MacDraw*). Students who cannot do their own writing can use software which combines text and graphics in many ways. For instance, they can dictate sentences to an adult or more skilled student, then go back and illustrate their story. This procedure encourages students to read and reread their own text.

Publication is a critical part of writing, and "desktop publishing" has found its way into many classrooms. When finished writing disappears into a student folder, the student's desk, or the wastebasket, the student has little sense that writing is for communication, not just for a grade. To enhance intrinsic motivation to write and write well, publication in some form—a class book, a newspaper, distribution to friends or parents, a periodic collection of writing for which each student selects his or her best work—is essential. Teachers stress that much of the excitement of a word processor comes from showing a professional-looking printed product to peers, parents, and even the principal. If the final product is enhanced with graphics and illustrations (either handmade or computer-generated), students are especially proud of their accomplishments.

Therefore, access to a printer is one of the key elements to successful word processing. Special education teachers may find themselves in the position of fighting for a share in the limited access to the school's printers. However, the professional look of a printed piece of writing that has been carefully edited and published using the word processor is a powerful motivator for reluctant writers. As one special educator put it, "My students are learning that their work can be more than just adequate; it can be excellent."

JUGGLING COMPUTERS, CLASSROOMS, AND STUDENTS' NEEDS

It is important to give students as much access as possible to the word processor. Many students lack confidence in their writing skills, and they need to explore the new writing tool without major distractions or time pressure. Because computers are a scarce resource in many schools, teachers need to find creative ways of allocating computer time. Teachers working in a resource room, who typically work with only a few students at a time, may want to give each student some time with the computer during a class period. If this strategy results in insufficient time for a student to work on a piece, it may be a better idea to schedule each student on the computer for one substantial chunk of time during the week. Giving students the opportunity to work on their own can have unexpected benefits, according to one resource room teacher. She sent individual students into the computer lab to work while she remained in the classroom with other students. She found that the students became quite independent and were proud of their ability to work on their own.

Teachers working with larger classes, including those working with special-needs students in a mainstreamed classroom, face problems in giving each student a turn at the computer. One solution is to have students take turns throughout the day working at the classroom computer, while other students engage in regular classroom activities. Research has shown that this approach is better than taking students to a computer lab, where they feel pressured to finish a writing task during a class period. Teachers who work in mainstream classrooms should be sure to give special-needs students plenty of time on the computer. In fact, some teachers advocate giving these students *more* time than their classmates. For students who have poor fine-motor control, for instance, this tool is more than just a welcome change; it may be critical to the development of their literacy skills. Students with learning disabilities often need the extra time, as well as extra encouragement, to translate their ideas to the screen.

Should students work alone, in pairs, or in larger groups? Although collaboration often initially comes about as a matter of necessity—too few computers for too many students—there are advantages in having students collaborate. As students learn to use the word processor, they can share their frustrations and successes with a partner. By working with peers who have particular strengths, students may develop new competencies, and the student "expert" feels accomplished in sharing his or her expertise. Pairing students with different strengths in generating ideas, illustrating, and editing may also help students make best use of their talents.

Teachers have noticed that students who collaborate become less dependent on the teacher and at the same time more able to critique their own and each other's writing (Russell, 1986). As a result, they take more responsibility for their writing. As Betty Church explained,

> Now I've started having the kids read to each other; when they've finished a piece of writing they read it to the group . . . the responsibility of the listener is to listen and respond to something they particularly like in the story. That's been great. I want the kids to interact a little more with each other rather than interacting with me on the writing. It's beginning to happen.

Donna Simone noticed changes in her students' proofreading skills as they began to collaborate:

> We have an activity once a week where the kids take one piece of their writing and they have to read it to their peer. In reading their work to one another they're picking up more of their errors and going back and doing the proofreading . . . . Somehow with the computers, we find that the reading part of it really helps. And it's not *me* saying, "Look, there's something wrong here. " Another kid is reading, and they *hear*, "That's not what I meant when I wrote it."

Students seem to enjoy reading each others' stories. Teachers find that they are much more interested in editing stories written by their friends than editing *canned* stories from a disk or a textbook.

Important as collaboration is, keep in mind that some students feel protective of their writing and are not yet willing to take the risk of sharing writing with another student or with the class. Students who view themselves as writing failures may need privacy in the beginning, and the word processor certainly can support this need. If at all possible, give each student his or her own disk for saving writing assignments. Encourage the class to respect these disks as personal property. While students should be encouraged to share their writing and to collaborate with others, at the same time their right to privacy must be respected. Teachers often make agreements with their students about which writing is private and which is to be published. For example, one group of students wrote lively and heartfelt essays about how they would change their school, but they agreed in advance that these would be published and shared only within their class, where they felt they could risk saying what they really thought about school rules, requirements, and activities.

STRATEGIES TO PROMOTE WRITING

As the student is working on the word processor, the teacher's role is critical in helping the child expand on his or her writing and in providing suggestions for doing so. Some teachers circulate throughout the room, acting as an unobtrusive assistant. Other teachers set a regular conference time to meet with each student to discuss the writing. This ensures that everyone gets a chance to get assistance, and students can learn to save their questions and comments for the assigned meeting time. Teachers report that successful word-processing interactions between student and teacher involve strategies like the following (Morocco & Neuman, 1986):

- Read what the student has written and react to it on a personal level. ("After reading that paragraph, I feel like I can almost taste that horrible meal you describe.")
- Help the child clarify or expand his writing by asking questions which directly relate to what the child has written. ("What is it about the room that makes it feel cheery?")
- Help the child plan what she is going to say, and review the plan with her while she is writing. ("You said you were going to concentrate on that nasty bee sting you got. How does this part about the picnic fit in?")
- Suggest strategies for expanding or clarifying what the child has written. ("Think back to when you went through the doors into the emergency room. Write about what was going through your mind.")
- Listen to what the child is saying, and ask her to write down just what she said. The child may also need help in remembering exactly what she said. ("That part about how you fooled your neighbor sounds very funny. Write down what you just said.")
- Type what the child is saying. The teacher is most likely to type for a student when ideas are flowing and the student is unable to type fast enough to get her ideas down, *or* when the student is stuck. Teachers can type short phrases on the computer, based on what the child says aloud, which the student can then expand into

complete sentences. This technique helps bridge the gap we so often see between the richness of a student's verbal account and the barrenness of the same account when it is written down.

- When the child encounters difficulties with the word processor, help her focus on the writing itself by assisting with particularly difficult commands or steps.
- Build the child's self-image as a writer by commenting on the strengths she has in common with real authors, and by assuring her that authors share some of her same frustrations.
- When you are working with a pair of children, demonstrate the skills described above, and encourage students to use the same strategies you are using when they are reading each other's work.

In sum, the teacher serves as audience, guide, and collaborator. The word processor helps the teacher by making the writing process more public and explicit and helps the student by providing clear and easily modifiable text, but teaching strategies that support and extend the child's writing are still the key to helping students become better writers.

DOES WORD PROCESSING REALLY MAKE A DIFFERENCE?

When teachers use word processors, they notice many changes in their students' approach to writing. The changes are not always easy to articulate, because they can occur in many different ways for different students and may include changes in attitude and self-confidence, which are difficult to measure. As Donna Simone explains,

> There are just so many things [the word processor] attacks; you can't say one is more important than the other. I used to think that this improved writing was really important, but I have to remind myself about the kid who wouldn't *do* the writing because he had such a lousy feeling, [and now] he's sitting right down when he comes in to do the writing. Also what's improving is the thinking going with the writing . . . . So it's like a whole spectrum of things that improve, and you can't say that one is more important than the other.

Teachers notice that special-needs students write more when they use word processors, that the quality of their word choice and sentences improves, and that students have improved images of themselves as writers. By keeping writing folders of students' work over the course of a year, teachers are able to document changes in writing skills that would not be apparent from examining only one or two assignments. Furthermore, writing folders enable teachers to show parents, principals, and the students themselves what has happened to the *quality* of writing over time. Teachers state that it is far easier to evaluate changes in students' writing when they have printed pieces from the word processor than when they have to struggle with a sheaf of handwritten pieces.

One researcher showed that when learning disabled students used a word processor along with a speech synthesizer, their writing improved in length, vocabulary, use of punctuation, and use of more complex writing structures (Rosegrant, 1985). The *Talking TextWriter,* based on Rosegrant's research, is now available commercially, as are other word processors that include speech synthesizers, such as *My Words.* Other researchers (e.g., Daiute, 1985) have found that when children write on the computer, the product is often a more natural, speechlike composition that has special meaning for the child.

However, one thing is clear from what we are learning about word processing from teachers and researchers: The power of word-processing software is quite dependent on the pedagogical skills of the teacher (Neuman & Morocco, 1985). Word-processing software is a tool that can facilitate writing, if used by a skilled and sensitive teacher. Providing students with a tool does not provide them with instruction nor an appropriate learning environment. It is up to the teacher to do this. It is gratifying to see that as teachers become more involved with word processors, the questions they ask start to focus more and more on *teaching writing* rather than on how to use the word processor. This is a good sign. It means that the tool is—as it is intended to be—an unobtrusive aid to the writing process. It means that we are correctly focusing our attention on helping special-needs students learn writing, rather than teaching them how to use the computer. As Betty Church said after using the word processor for nearly a year, "I've shifted gears from focusing on disabilities to focusing on real writing."

NOTES

Teachers who want to use word processors can choose from a number of available programs but will most likely begin with the program already in use at their schools, unless it is clearly too simple or too complex for their students. For teachers who are initiating word processing in their schools, we suggest they not only spend time previewing programs but also talk with others who are using programs which they may want to purchase. Remember that word processors that may take longer to learn may also include more options, such as a thesaurus. If a teacher believes these options will benefit his or her students, then increased complexity or learning time may be justified. *Magic Slate, Talking TextWriter, Bank Street Writer III, Milliken Word Processor, FS:Write, AppleWorks,* and *FrEdWriter* are among the numerous word processors appropriate for school use.

Teachers who incorporate strategies to promote writing in their writing lessons may want to explore pre-writing software programs, such as *Proteus, Write Connection,* or *Write On!* Many programs allow students to transfer their notes to their word processor to continue their writing.

REFERENCES

Daiute, C. (1985). *Writing and computers.* Reading, MA: Addison-Wesley.

Graves, D. (1983). *Writing: Teachers and children at work.* Portsmouth, NH: Heinemann Educational Books.

Graves, D., & Calkins, L. (1980). Research update: When children want to punctuate: Basic skills belong in context. *Language Arts, 57*(5), 567–573.

Morocco, C., & Neuman, S. (1986). Word processors and the acquisition of writing strategies. *Journal of Learning Disabilities, 19,* 243–247.

Neuman, S., & Morocco, C. (1985). A model teaching environment for using word processors with LD children. *The Writing Project Technical Report No. 2.* Newton, MA: Education Development Center.

Rosegrant, T. (1985). Using the microcomputer as a tool for learning to read and write. *Journal of Learning Disabilities, 18,* 113–115.

Russell, S. J. (1986). *Creating an environment for change in the teaching of writing.* Paper presented at the annual meeting of the American Educational Research Association, San Francisco.

ADDITIONAL RESOURCES

Calkins, L. (1986). *The art of teaching writing.* Portsmouth, NH: Heinemann Educational Books.

Collins, J. L., & Sommers, E. A. (1985). *Writing on line: Using computers in the teaching of writing.* Upper Montclair, NJ: Brynton/Cook.

Eiser, L. (1986). I luv to write! Spelling checkers in the classroom. *Classroom Computer Learning, 7*(3), 50–57.

Eiser, L. (1987). No more hunt-and-peck! *Classroom Computer Learning, 8,* 52–56.

Garvey, I. (1986). Spelling checkers: can they actually teach spelling? *Classroom Computer Learning, 5*(4), 62–65.

Hillocks, G., Jr. (1987). Synthesis of research on teaching writing. *Educational Leadership, 44,* 71–82.

King, D. H. (1986). *Keyboarding skills: All grades*. Cambridge, MA: Educators Publishing Service.

McCarthy, R. (1988). Stop the presses! An update on desktop publishing. *Electronic Learning, 7*(6), 62–65.

Parham, C. (1986). Conquering the dreaded blank page. *Classroom Computer Learning, 1*(1), 39–42, 44.

Riel, M. (1985). The computer chronicles newswire: A functional learning environment for acquiring literacy skills. *Journal of Educational Computing, 1*(3), 317–337.

4. Using the Computer to Develop Problem-Solving and Critical Thinking Skills

The computer offers new approaches, activities, and tools for special education students in the familiar curriculum areas of reading, writing, mathematics, science, and social studies. But some of what the computer has to offer neither relates to the development of familiar skills nor fits neatly into traditional subject matter categories.

In this chapter we take a look at computer software designed to help students *develop* and *use* the ability to cope with unfamiliar problem situations, situations which approach more nearly the complexity and unpredictability of real life than do most textbook problems. This software is not always easily categorized in terms of the usual sort of learning objectives. At first glance, it may appear difficult to justify in a tightly scheduled, highly accountable regular or special education setting. However, software that helps students learn how to solve problems offers them opportunities to develop confidence and skill in decision making and critical thinking—processes that we know to be of great importance to their survival beyond school.

The software we discuss in this chapter covers a broad range of content and complexity. At one end of the spectrum, some of this software offers straightforward problem-solving games which involve the student in discovering a secret pattern, sequence, or relationship, usually by trying something, seeing the result, then trying again. For instance, in *The Pond*, students try to find a pattern of moves (e.g., down 5, left 2, up 3) which, when repeated a number of times, will enable the frog to jump successfully through a maze of lily pads from one end of the pond to the other. Such a game may require only a few minutes for completion of one puzzle.

Other problem-solving software is much more elaborate and time-consuming. In using a simulation, for example, the student may participate in a story which has a setting, a goal, perhaps other characters, objects to use, events which occur, and rules which may be stated or which may be left to the user to uncover. By making decisions and accumulating information, the student can gradually reach the goal, which may be to solve a mystery, to find something, to get to a particular place, or to prevent a disaster. For instance, in *Snooper Troops*, the student is a detective who must find clues and interview people in an attempt to solve a mystery. This problem may take many sessions and may require the cooperation of several students to complete.

While you are reading through this chapter, keep in mind the following thoughts:

- Problem-solving software includes a very broad range of content and complexity.
- Any particular piece of problem-solving software can be appropriate for a variety of ages and abilities.
- Student cooperation can allow students with a range of strengths and needs to work together in using this software.
- Special-needs students' ability to cope with this software has often surprised their teachers.
- Watching special-needs students work in these less familiar contexts has helped teachers understand more about their learning styles and strategies.

This chapter includes examples of how special-needs teachers have used problem-solving software in different ways, the learning objectives they have developed, and how they see their own role in supporting students' use of this software.

"BUT MY STUDENTS COULD NEVER DO THAT!": WHY USE PROBLEM-SOLVING SOFTWARE?

In addition to the four general characteristics of learner-centered software described in Chapter 1 (user control over the goal or strategy or both; informational feedback; the use of prediction and successive approximation; meaningful context), problem-solving software has all or most of the following characteristics:

1. It presents unfamiliar problems in unfamiliar contexts.
2. While students may need to draw on some skills that fall into conventional subject matter areas (e.g., estimation, map reading), much of what is required crosses subject area boundaries (e.g., sequencing, testing hypotheses, revising strategies).
3. It requires (and encourages) risk taking and initiation.
4. Errors are necessary and unavoidable.

5. Information is often given in more than one mode. Information is offered visually as well as through text or symbols.
6. In order to solve problems, many facets of the situation must be considered and coordinated. Depending on the particular problem, synthesizing a very large amount of information and paying attention to many variables may be necessary.
7. Directions are of limited usefulness. Students find out most about the problem situation by trying something and seeing what happens.

You may already be starting to say, "No directions? Many variables? Risk taking? Complex problems? My students could never do that!" Many of us reacted this way when we first encountered problem-solving software. It intrigued us as a potentially interesting and useful learning experience, and we wished we could make more such experiences available to our students. However, we were all too aware of the difficulty many of our students seem to have with material that is not presented carefully, slowly, sequentially. However, as teachers began to try such software, they found that many students functioned differently in this new context than in their regular school work. Teachers learned new things about their students' strengths and learning styles, while the students themselves found new ways of working on their learning problems. In some cases, teachers found that working with students in the computer problem-solving environment led to a much improved student-teacher relationship in other aspects of classroom work. You will encounter some of these firsthand experiences later in the chapter.

Some of the characteristics of problem-solving software are actually advantageous for some special-needs students. Students with different language, visual, and reasoning strengths can often be successfully paired to work on these activities because of the range of modes in which information is presented. A child who is a poor reader but is quick to see visual patterns may excel in a problem-solving game which emphasizes classification of graphic images. Lack of directions may encourage children who have always failed at following written directions; the necessity to try something and see what happens brings about a certain amount of equality among students—*nobody* knows, at first, what to do!

There are also practical, educational reasons for beginning to include problem-solving computer activities in our curriculum. Lack of directions, complexity, and the inevitability of making errors—these are the conditions of solving problems in daily life. Looking for a job, planning a trip, buying a used car, or managing a budget are activities for which there are no directions and no quick solutions. In fact, many of these problems are never really "solved" at all; they require continual reevaluation and planning.

Providing" real life" problem-solving experiences in an educational setting, without watering them down so far that they lose exactly those characteristics of reality that we are after, has always been a difficult task. Many teachers have attempted to involve their students in the kinds of complex projects which are optimal for the development of thinking, reasoning, and organizational skills. Such projects—running a school store, planning and cooking a meal, publishing a newspaper, writing and presenting a play—allow students with many different kinds of learning problems and learning strengths to participate successfully. Often these activities are a vehicle by which special-needs students find a role different from their usual one of failure. Such projects happen rarely for most students, although recent curriculum efforts in mathematics show some promising directions (e.g., the TERC Used Numbers project). Problem-solving software gives us a new way to offer students a greater variety of experiences with unfamiliar problems and more opportunities to develop the skills necessary to solve them.

THREE PROGRAMS THAT WORKED

The best way to give you a sense of how problem-solving software can be used with special-needs students is to share the accounts of several teachers who, each in her or his own way, developed objectives, approaches, and activities which worked with a particular student population. These classroom stories are not intended as prescriptions. The teachers themselves would be the last people to offer their experiences as the only way, or even the right way, to use this software. All of them are continuing to develop and modify their approaches. These stories simply provide glimpses of the possibilities, and we hope you will find flashes of the familiar in encountering these classrooms.

Snooper Troops

Snooper Troops is a simulated mystery story in which the user plays the role of detective. Students must travel around the town and collect clues by interviewing various characters in the story. Jan Schraitle chose *Snooper Troops* to use with her class of 12 seventh-grade learning disabled students in order to focus on student improvement in six areas: map skills, organizational skills (such as note taking), deductive reasoning, communication skills, cooperation with peers, and the ability to work on a project not completed within one class period.

The students were at least two grade levels behind in reading as well as two to three grade levels behind in written expression skills. The class was in the computer lab 2 days a week to use the word processor and on one of these days, students used *Snooper Troops* on a rotating basis. (They had three copies available, so this limited the number of students who could work on it, since the disk must remain in the disk drive during use.)

Jan introduced the whole class to the mechanics of operating the program, then chose partners based on particular objectives she had for work on interpersonal skills. Students began by practicing the mechanics of the program, such as driving the Snoopmobile. Students at first had difficulty thinking ahead in order to press the command to stop in time. To a chorus of "oops," and "oh, no, not again," students gradually learned to control the car. At first, Jan gave no directions for proceeding to solve the case, but as students worked, it became clear, both to her and to the students, that further direction was needed. She devised a worksheet to help students organize information from the booklet which accompanies the program and an outline map on which students could fill in the places they located as they investigated. She met with each group at "frustration times," directing the students through questioning, in order to guide them in determining next steps. At intervals, three pairs would meet as a group to compare notes and learn from each other.

In working with this game, students learned—of necessity—to keep good notes, to read information in the booklet, and to use their maps. Students also learned that they could not solve the case simply by guessing. While this was frustrating at times, they learned that they had to gather facts to back up their suspicions. Students loved this game, and those who solved the mystery were secretive and amused by others' guesses. Jan saw particular improvement in organization, communication, and cooperation among her students in the course of working with this software.

Agent U.S.A.

Agent U.S.A. challenges students to locate and neutralize the "fuzz-bomb." In order to do so successfully, students travel on trains around the United States searching for the fuzz-bomb, a search which requires attention to both time and geography. Steve Spencer used this program with a self-contained class of 10 students, grades 4 to 6, with reading or language problems and behavior problems directly related to their learning difficulties. Steve's goals included organizational skills, helping students to expand their strategies beyond a trial-and-error approach, reducing antisocial behavior, and improving students' self-images as learners.

Steve began with a short introduction to *Agent* and almost immediately began playing the game and having students join in. Because his students have difficulty listening to a long stream of information, he felt that the best approach was to involve them in trying the game immediately, then gradually add information through discussion. When the students were initially unsuccessful, he helped the group discuss possible strategies.

Steve found that a group of three students at the computer was the most effective grouping. One student studied the map, one typed, and the third helped out with strategies. He often included in the group one student with severe reading problems, but average conceptual skills, and one student who was a better reader with poorer cognitive skills.

One group of three 12-year-old boys became particularly involved with *Agent*. They saw it as different from the usual school tasks and were surprised that Steve allowed them to play it frequently. As they worked with the game, the group began discussing and trying new strategies and establishing long-term goals. Steve noticed increased skills, concentration, and cooperation. One boy in particular, Allen, with a severe reading disability and visual perceptual, auditory, and memory problems, had refused to try anything and was becoming a troublemaker. He was turned off to everything, and, as Steve says, "if you can't solve that problem, you can't teach." But like many learning disabled students, Allen was smart and was able to develop effective strategies in this context, one that appealed to him and in which he felt more in control. The experience boosted his view of himself, improved his attitude, and helped him develop and use some important organizing strategies.

Steve felt that this activity required a lot of teacher supervision. He checked in with his students often while they used the game, helping them to consider new ideas and strategies. In the future, he thinks a group discussion after each session to review what occurred and develop strategies for the next round would be useful. An interesting footnote to Steve's experience is that none of his students actually solved the game. However, Steve's students found immense satisfaction and success in getting better and better at a challenging, demanding activity. We often protect special-needs students from frustration and failure by ensuring that they gain immediate success, yet frustration is a necessary component of the learning process. In real life, success is often measured in years rather than minutes. This kind of experience with a long-term problem may help students develop the kind of concentration and involvement they will need when tasks do not begin and end within a 45-minute period.

Gertrude's Secrets and *Gertrude's Puzzles*

Gertrude's Secrets and *Gertrude's Puzzles* each contain a series of games which require sorting and classification by shape and color. For example in one of the puzzles, the user must arrange puzzle pieces in three rows and three columns so that no piece is in the same row or column as another piece of the same shape or color. Many special education teachers have used these programs with a range of populations. Linda Ware uses both *Secrets* (the easier of the two programs) and *Puzzles* with her junior-high resource class of students with learning and emotional problems. Her objectives in using this software are to promote her students' ability to discriminate between color and shape, order and classify by pattern and by sequence, categorize and infer patterns and rules, recognize similarities and differences, use deductive reasoning, and use critical thinking.

She finds that her students are eager to use the computer, considering it "play" rather than "work," but that their responses clearly indicate that they are thinking not only about selecting an answer but about *why* they are selecting a particular response. She organizes her class into groups of three, introduces the computer work to one group, then appoints a Computer Tutor (CT) from the first group to help the next group of three get started. This kind of organization enables her to work with students not involved with the computer.

Linda has been particularly interested in developing off-computer activities to give students the chance to use and extend the learning she sees going on while her students are using the computer. She creates worksheets that guide students to use the same kinds of comparison and categorization skills with which they have been engaged on the computer in other areas of the curriculum. In one activity, students worked on the similarities and differences among animal, plant, and human cells from their science notes; in another, they classified bodies of water in their state. Huge arguments arose from their work on classifying rock musicians. Linda reports that their classification scheme was complex and entirely student-directed: "Throughout their work, they immersed themselves in the data in a manner which rarely occurs in the classroom; they were thinking about the data rather than merely memorizing it without meaning-making." With her support, she felt that there was an obvious transfer of skills learned while using *Gertrude* to other content areas.

Steve Voiles has used *Gertrude's Secrets* with everyone from his own 5-year-old to his sixth-grade special education students. He finds that these learning activities seem to stimulate the same intense interest and perseverance throughout this broad age range. Like Linda, he is interested in learning objectives that include hypothesis and prediction, pattern recognition, and using deductive reasoning; he also modifies his approach to fit the emotional and social needs of his students. Here are Steve's accounts of two students' experiences with *Gertrude*.

> Jeffrey was extremely defensive with teachers, especially special education teachers, because he had been teased about being a "dummy." Part of him was valiantly trying to prove that he was not dumb, but his emotions were so overwrought that he would make quick, poorly-thought-out decisions based only on partial information. It was extremely hard to instruct him because he was forever trying to prove that he didn't need any help, and his rejection of assistance placed him in an even deeper morass of partial understanding and poorly concealed self-doubt. The attractive format of *Gertrude*, however, momentarily disarmed him. When I allowed him to "play," he faced a computer, not a teacher; his defensiveness slowly began to fall away. After some preliminary success, I began to offer observations about the choices he was making. I was a commentator more than a teacher, so I was tolerated. After a couple of sessions, Jeff began to be willing to speculate, hypothesize, and then test his ideas to see if he was right. He was thinking carefully instead of reacting emotionally and impulsively. His intelligence began to show through and our whole teacher-student relationship began to improve.
>
> Kevin, however, baffled me with his hyperactive approach to the activities. He was bright, but impulsive. The keying caused him no problem and the maze was a breeze for him. When he got to an activity, his fingers would fly over the keys as he systematically inserted piece after piece into the puzzle until a solution was reached. He knew that trial and error would eventually solve the puzzle, and he was in a hurry to pile up prizes. I was disappointed to see him settle for a primitive strategy when he was so clearly capable of higher levels of thinking. Yet, when I tried to engage him in hypothesis and prediction, he just saw it as slowing him down and "making things harder." I put the disk away for a while, not wanting to help him overdevelop such a low-level skill while ultimately more satisfying approaches lay untapped. Finally I realized that I could exploit his competitiveness as a means to get him to pay more attention to details and options. I had Kevin work with a partner, taking turns at the keyboard. While Kevin solved a puzzle, his partner would count the number of guesses that he needed to solve it. Then they would switch positions and Kevin would count the number of guesses required for his partner to

solve the same puzzle. Very quickly, they began to compete. I innocently pointed out that, *if they were competing*, the winner would be the person who could solve the puzzle in the fewest possible guesses. Suddenly Kevin was willing to listen to suggestions and explore approaches that might lower his score.

For all of these teachers, effective use of problem-solving software required extra time, effort, and thought. All of them had to make the activities work for their particular group of students by designing support materials, deciding on grouping, or choosing how and when to intervene and when to hold back from intervening. For these teachers, the computer was not viewed as a way to save time or to make learning more efficient. Rather, it offered them and their students a flexible new approach for achieving important learning objectives. We consider these objectives further in the next section.

LEARNING OBJECTIVES FOR PROBLEM-SOLVING SOFTWARE

What are appropriate learning goals and objectives for problem-solving software that does not fit neatly into a content-area category such as mathematics or language arts? Teachers using this software have had to grapple with this issue, partly to explain to others—parents, administrators, next year's teacher—how and why this software is used, but also to help themselves clarify objectives and plan for individual students.

Here, for example, is what two teachers say about their objectives in using such software. Steve Voiles describes the use of *Gertrude's Secrets* this way:

> I don't believe *Gertrude* directly fits a standard curriculum area, but I value it as pure "cognitive exercise." You have to think to explore the maze and the activities. If students are encouraged to develop strategies and to state their ideas about why one possibility works and another does not, then several additional layers of mental exercise accumulate. Depending upon the student and his particular level of ability and experience, you might choose to focus on deductive reasoning, hypothesis and prediction, sets and subsets, shape and color recognition, the process of elimination, trial and error, pattern recognition, etc.

Steve Spencer explains the reasons for using *Agent U.S.A.* in his classroom:

> There are so many things my kids don't know. Their basic problem is a lack of organizational ability. They can't get

organized to attack a problem. Instead they use trial and error only, get frustrated, and give up. I work on this in all areas, and *Agent* was one fun way to do it. The kids need help in a tremendously wide area—both academic and social. How do you teach concentration? You can't. You can't teach it explicitly. You try to help kids begin to ask themselves questions, develop strategies, and look at things in different ways.

From these statements, you can see that teachers' goals for their students' use of problem-solving software can cover a broad range of objectives. Many teachers have begun by trying out problem-solving software with their students without preconceived ideas about goals and objectives, to see what the potential uses are for their particular group of students. Teachers find they can't always predict how their students are going to interact with problem-solving situations. However, after some experience with a piece of software, teachers usually devise more specific goals. They translate the general objectives they started out with, objectives such as concentration, organization, or "cognitive exercise," into more specific, more manageable goals that can be included in children's educational plans and which lend themselves to monitoring and documentation. Depending on the constraints or flexibility in their particular setting, their own teaching styles, the strengths and needs of their students, and the curriculum for which they are responsible, teachers may choose to concentrate on social skills, general learning skills, or specific content area skills. More and more schools are including objectives in critical thinking or problem solving in their curriculum for all students; teachers' uses of problem-solving software with their special education students often match such objectives extremely well (e.g., Pogrow, 1988).

What follows is a selection of objectives based on those used by our contributing teachers. We offer this list as a beginning which we encourage you to examine, select from, expand, and alter appropriately for your particular group of students. Since some of these objectives were suggested by teachers at many grade levels, and since strengths and needs of students can vary so widely at each grade level, we have categorized them by type of objective only, not by age or grade.

A SAMPLING OF PROBLEM-SOLVING GOALS

(Remember! This is a sampling of goals used by special education teachers, not a complete or definitive listing.)

Organizational Skills

- Note taking
- Gathering facts
- Categorizing
- Comparing and contrasting
- Creating and using organized lists
- Identifying patterns
- Sorting necessary and unnecessary information

Reasoning Skills

- Deductive reasoning
- Finding multiple solutions
- Constructing a sequence of events
- Modifying a sequence of events
- Reasoning backwards from a result to the sequence which led to it
- Using trial and error effectively
- Moving from sole use of trial and error to a range of other strategies
- Using a process of elimination to isolate the solution
- Solving problems with minimal clues
- Varying one aspect of a situation at a time to isolate critical attributes
- Evaluating partial solutions
- Testing solutions
- Making sense out of contradictory or ambiguous information
- Evaluating relative importance of different elements in a situation

Learning to Learn (See Chapter 5 for more about this topic.)

- Working on a project not completed in one class period
- Responding to situations flexibly
- Concentrating on a task
- Learning to tolerate errors
- Controlling impulsive answers
- Using errors as information to guide next steps
- Sticking to a goal

Social Skills

- Cooperating with a peer or small group
- Communicating with peers about content and strategy
- Taking turns
- Becoming a "student expert" or "computer tutor"
- Taking a leadership role

Content Area Skills

- Working with maps
- Improving language development
- Recognizing shapes and colors

STARTING OUT WITH PROBLEM-SOLVING SOFTWARE

While we cannot imagine a checklist or set of rules which would adequately guide the selection of problem-solving software, much less appropriately match it to students, we have identified four guidelines which are important to consider when selecting and using software for work on critical thinking skills.

First, most teachers advocate beginning with one piece of problem-solving software and exploring it thoroughly with students. Problem-solving software is typically time-consuming and complex. Students need enough time to familiarize themselves with both the mechanics and the ideas of the software before they can focus on the problems themselves. As indicated by the accounts in this chapter, a good piece of problem-solving software can provide many sessions of productive work at the computer as well as class discussions and noncomputer activities. Teachers, too, need time to make connections between the computer experiences and other parts of their curriculum. Extended, thorough use of a single piece of software appears to lead to a more productive, integrated, and coherent experience.

Second, select problem-solving software that offers the student a small world consisting of a setting and (usually) characters which create a believable context for the problem. The story context need not be complicated and detailed to be effective in engaging students in solving problems. Much simpler worlds, such as the frog-pond context of *The Pond*, are intriguing and attractive, even for older students. By "believable" we do not mean realistic; rather, we mean that the problem emerges naturally and is clearly related to the context which is developed in the software. For example, in many pieces of software the user travels through a maze of interconnected rooms, encountering hazards and acquiring treasures. If in order to enter the treasure room in a magician's castle, you have to use clues you have gathered to identify the magic words which open the door, this activity is perfectly consistent and natural, given the premise of the situation. However, if you are asked to solve an arithmetic problem each time you want to move to a new room, the situation is contrived to lure you into mathematics practice, and any student can tell the difference. Settings and characters which are purely decorative but do not have a function in the problem situation appear to be of much less interest to students over time.

Third, problem-solving software in which trial and error is a possible strategy, but not the only or the best strategy, seems to work well with many different students. The opportunity to begin by using trial and error offers a low-risk entrance into problem-solving software. No one can solve problems such as these without some experience. Trying without succeeding, then trying again and again until some success is achieved, is a skill—and requires an attitude—which many students must learn in an environment which appears to them a safe one in which to take such a risk. For many children, some success with trial-and-error lays the foundation for gradually moving toward more sophisticated reasoning strategies. For others who are masters of trial and error, software in which trial and error is not sufficient to solve the problem may encourage them to adopt more sophisticated strategies. Software which allows this strategy but makes it worthwhile for students to devise more sophisticated strategies counteracts, on the one hand, too much initial frustration and, on the other hand, boredom resulting from lack of challenge.

Finally, match the complexity and duration of problem-solving activities with students' levels of experience with this kind of software. Students who lack confidence in their own intellectual ability, who are frightened by unfamiliar learning situations, or who have poor organizational and reasoning strategies will need to enter this new realm with appropriate support and structure. A good place to start is with software that limits the number of choices students must make, has relatively few steps to reach a solution, has a manageable amount of information for students to collect and organize, offers on-screen prompts about next steps, and is short enough to be completed in a single sitting. For example, *Where in the World is Carmen Sandiego?* has all these features and provides a straightforward, but still challenging, introduction to this whole genre of mystery simulations.

By choosing appropriate software we can help students gradually become more confident and independent in problem solving, but we should not expect to eliminate all the difficulties students may encounter. Teachers have found that they need to be wary of overprotecting their students. We don't want students to fail, yet experiencing some degree of failure is a component of solving problems. If a student is ever going to be able to work on a problem which cannot be completed in one session, learn how to work cooperatively with a peer, or manage a period of frustration, he or she must *have* these experiences.

The role of the teacher is a critical part of this process. What is clear from teachers' accounts of their use of problem-solving software with their special-needs students is that successful incorporation of this software into the learning environment requires a triad: student, software, and teacher. It is just as unreasonable to expect students' critical thinking skills to blossom automatically when they use a piece of software designed to encourage problem solving as it is to assume students will learn how to read if given enough books. Books provide motivating and intriguing content—a reason for reading. Problem-solving software also provides motivating and intriguing content—a reason for thinking and planning. If students are to make the best use of this software, teachers are there to help students over mechanical hurdles, provide support during frustration, encourage productive failure as well as success, and extend new learning into other aspects of the students' work.

NOTES

Students and teachers who enjoy *Where in the World is Carmen Sandiego?* may relish the challenge of other programs in the series: *Where in the U.S.A. is Carmen Sandiego?* and *Where in Europe is Carmen Sandiego?* These programs can mesh effectively with off-computer activities in areas such as mapping skills and reference tools. *The Enchanted Forest, Factory,* and *Gears* are a few other examples of problem-solving software for the classroom.

REFERENCES

Pogrow, S. (1988). HOTS: A thinking skills program for at-risk students. *Principal, 67*(4), 19–24.

Used numbers: Collecting and analyzing real data. Mathematics curriculum modules funded by the National Science Foundation, Grant #MDR8651649. This project is under development. For more information, contact Technical Education Research Centers, 1696 Mass. Ave., Cambridge, MA 02138.

ADDITIONAL RESOURCES

Malone, T. W. (1980). *What makes things fun to learn: A study of intrinsically motivating computer games*. Palo Alto, CA: Palo Alto Research Center. Xerox.

Meltzer, L. J. (1984). Cognitive assessment and the diagnosis of learning problems. In M. D. Levine & P. Satz (Eds.), *Middle childhood: Development and dysfunction* (pp. 131–152). Baltimore: University Park Press.

Meltzer, L. J., Solomon, B., & Fenton, T. (1987). *Problem-solving strategies in children with and without learning disabilities*. Paper presented at the 95th annual convention of the American Psychological Association, New York.

Weir, Sylvia (1987). *Cultivating minds: A Logo casebook*. New York: Harper & Row.

5. Learning to Be a Learner: Using the Computer to Improve Motivation, Responsibility, Perseverance, and Insight

All learners are expected to acquire knowledge and skills in a variety of areas during their school careers: reading, writing, mathematics, science, social studies, consumer skills, study skills, and so forth. But there is one special kind of knowledge that students, including special-needs learners, acquire whether or not it is taught to them explicitly. *They learn what learning is.*

The acquisition of knowledge about learning and about themselves as learners is a critical part of students' education. Students are learning about learning all the time, directly and indirectly. Messages about learning may be subtle, yet students read those messages clearly, perhaps more clearly than they read their current assignment!

Consider a student making a consistently high number of incorrect responses while using a piece of software. She is automatically moved to lower and lower levels of difficulty until she reaches the lowest level. When she continues to perform poorly, a message appears on the screen: "Stop fooling around. I know you can do better than this." This real incident taught the student much more about herself as a learner than it did about the subject matter it was designed to convey. She learned that she was unbelievably incapable and, even worse, that there was no hope and no help for her to improve. (Her teacher, who had not explored the software down to its lowest level, was also shocked and dismayed about this negative message.)

Although there is still software on the market—some of it labeled "special education software"—which presents users with sad faces and other negative responses, most software does not include such obviously destructive messages. However, almost all software gives some kind of indication to the student about the learning process through its goals and difficulty levels, its motivational features, the kind of help it provides (or does not provide), and the feedback it gives students about their attempts at the task.

Because the computer is interactive, with its own demands, feedback, and messages to the user, it is potentially both more useful and more dangerous than other media for teaching students about learning. This chapter examines what students learn about learning when they use software and what features of software make a difference in students' views of themselves as learners.

GOALS FOR CHILDREN: LEARNING HOW TO LEARN

Before we examine the features of software that contribute to students' views of the learning process, we need to consider what it is that we do want students to learn about learning and about themselves as learners. While students leave school with some portion of the critical knowledge and skills they will need in later years, they can never learn everything they need to know to solve problems throughout their lives, at work, in their families, and in their communities. After they leave school, they will often need to learn new skills, weigh options and make choices, and apply their skills in unfamiliar situations. If students come out of school with only a particular set of skills in reading and mathematics, we have failed them. They must also understand how to approach a new learning situation.

The qualities we want students to develop for lifelong learning transcend particular concepts and skills. These include:

1. Motivation: the desire to learn
2. Perseverance: the willingness to continue to learn through confusion or frustration
3. Responsibility: the monitoring and evaluation of learning
4. Insight: understanding one's own learning strengths and needs

In the following sections of this chapter, we describe each of these qualities and then explore software features which can support or undermine their development.

Motivation: Beginning the Learning Process. Without the desire to learn, most students learn little. As one of our contributing teachers says about students' lack of motivation, "If you can't solve that problem, you can't teach." Researchers and teachers have long recognized motivation as a critical part of the learning process (e.g., Adelman & Taylor, 1977; Paris, Lipson, & Wixson, 1983). Motivation is a necessary, although not sufficient, condition for real learning. Of course, the motivated student is not necessarily engaged in learning—you can think of many things your students are very motivated to do which you might not consider educational! Nevertheless, without motivation, learning will either not occur at all or will soon be forgotten.

Perseverance: Continuing the Learning Process. Many students who have begun to learn something new soon become discouraged because they do not understand the nature of learning. Many students believe that good learners always know the answer right away and that any hesitation or confusion in solving a problem is an indication of lack of ability. In fact, unsuccessful learners often do not distinguish between *learning* and *knowing*. Students need to learn that the learning process often includes a period of frustration and confusion and that good learners make many false starts, try many strategies, and revise their work often. They must understand that making errors and being dissatisfied with one's work initially is inherent in the process of attempting to do something which one has not done before.

Learners who know that they can work towards success and that they will often not succeed immediately, but only after devoting considerable time and effort to a task, have a "mastery" orientation to learning. The mastery-oriented learner believes that, in a new learning situation, effort will lead to success, and that the learner has a great deal of control over the ultimate outcome. In contrast, "helpless" learners believe that learning is not within their control, that success is most often due to luck rather than to effort, that no matter how hard they try, they are bound to fail. Researchers have shown that many special-needs students exemplify this "learned helplessness," the symptoms of which—passivity, avoidance, defiance, anger—are all too familiar to their teachers (Pearl, 1982; Pearl, Bryan, & Donahue, 1980). The student who can never find his or her pencil, who asks after every problem, "Is that right?", or who gives up after the first mistake, is a "helpless" learner. These learners may begin a task but rarely persevere without a nearby adult "pulling teeth."

Responsibility: Monitoring the Learning Process. One of the "learning to learn" skills most often lacking in special education students is consistent, careful self-monitoring. One aspect of self-monitoring is evaluation of progress during learning—thinking about why errors may have occurred, revising strategies or goals, scrutinizing quality, assessing how close one is to the desired outcome. Taking responsibility for learning also involves setting one's own goals and determining the criteria for reaching them. In their post-school lives, students will sometimes need to respond to tasks and criteria set by others, but they will often have to determine for themselves when what they have done is "good enough." Responsible learning requires acceptance of final outcomes, both failures and successes. While "helpless" learners may see failures as inevitable and successes as due to chance, mastery learners accept both failure and success, learning from each what works and what requires improvement (e.g., Licht & Dweck, 1984).

Insight: Understanding One's Own Learning Strengths and Needs. Adults often recognize what works best for them in a learning situation. Some of us know we need a map to find our way to a new place; others need a careful sequence of written directions; still others would rather rely on landmarks and visual cues. Because special-needs learners have problems with learning, it is particularly important for them to know how they learn best, what kinds of information it is easiest for them to process, how they can transform information to make it easier to handle, and which of the characteristics of their approach sometimes impede learning. They need to know what helps them remember or retrieve previous learning, understand directions, or approach a new problem. They need to know when their preferred style of problem solving is likely to work and when it gets in their way. They need to know what is difficult for them, what strengths they can draw on, and what external aids are useful to them.

If, when students finish their formal education, they have the desire to learn, they understand what the learning process is likely to entail, they take responsibility for monitoring and evaluating their own progress, and they know their own strengths and needs, they indeed have the skills to approach successfully whatever new learning situations they may encounter. With these "learning to learn" goals in mind, we turn to the features of software which can help or hinder the progress of students toward becoming motivated, persevering, responsible, and insighthful learners.

SOFTWARE FEATURES THAT AFFECT LEARNING HOW TO LEARN

A major strength of learner-centered software, when carefully chosen and used in the appropriate context, is that it can help students learn to become better learners. In addition to helping them master the subject area, this software helps them learn what their capabilities are, that it is possible to learn even when initially frustrated or confused, that asking for help or feedback is an important way of learning, and that many different kinds of learning strategies can be effective in reaching a goal. The four major characteristics of learner-centered software—learner choice, informational feedback, prediction and successive approximation, and a meaningful context—emerged from teachers' concerns about their students' difficulties in learning how to learn. While these characteristics provide general guidelines for selecting software that encourages students to be learners, many specific software features affect children's attitude and engagement. Critical software features include those that affect:

1. The way the task is motivated
2. The balance between student control and software control of the learning process
3. The messages the student receives about progress

Intrinsic Motivation: Software Features Affecting Interest and Engagement

Teachers are the first to recognize that certain pieces of software facilitate learning by helping students take the important first step of attending to the task. When thinking more about "motivational features," though, we are actually considering an important aspect of learning that goes beyond simply attending to the problem. The motivation that interests us as educators is based on the *intrinsic* desire to learn, a desire that is present in everyone. While software alone cannot generate intrinsic motivation, when combined with thoughtful teacher interventions and an appropriate learning environment, computer experiences can go a long way toward engaging students in the learning process.

Some software captures students' attention through the use of special effects ("bells and whistles") that have little to do with the learning task. A tune is played each time the child scores a point or colorful graphics are used to attract students to a task. Graphics and sound certainly have some attention-getting value, and can be used effectively to draw in a child who is reluctant to get involved in a new learning situation. But there are some built-in limitations to these features. Initially, students may be entertained by the materialization of a monster on the screen each time a certain move is made, but after they've seen it a few times, the repetition of this graphic may actually result in the child's losing interest, *unless the task itself is engaging and motivating as well.* Research supports teachers' observations that after a period of time, most students are less interested in these special effects and instead pay more attention to the features that have some relevance to the learning task (Lepper, 1985). If the learning task itself is boring, inappropriate, or ineffective, special effects alone will not engage children in *learning*, even if they remain attentive to the software. Intrinsic motivation related to children's sense of their own mastery has consistently been shown to be more effective for promoting learning than external rewards which are based on an outside judge's (in this case, the computer's) rating of student work.

Teachers should be cautious in using software that has many "bell and whistle" features for another reason, as well. A potential danger of prominent but irrelevant sound or graphics is distraction from the salient features of the learning task. The problem of distraction can be a significant barrier during the crucial early phases of learning, when the child is attempting to figure out which dimensions of the task are important. Research tells us that children with learning disabilities are particularly likely to be distracted by irrelevant detail (Meltzer, 1984). Given software with many attractive yet unimportant features, students may have a hard time figuring out how to distinguish the central problem from the background. Rather than avoiding all software with bells and whistles, we need to be alert to circumstances in which these features become an attractive nuisance that interferes with learning, and those in which they can be helpful.

If graphics and sound do not necessarily lead to intrinsic motivation, what software features *can* motivate a child to learn? There is a great deal of research about what characteristics of a task lead to intrinsic motivation to learn (Lepper, 1985; Malone, 1981). These include factors such as an appropriate level of challenge, environments that appeal to human curiosity through surprise and variation, high amounts of choice, clear relationships between actions and consequences, and the connection of learning to the "real" world. These factors suggest that the context of the learning task is important. Many excellent software programs draw us in by creating an absorbing context, much as a good novel sets a scene that is immediately engaging. Sometimes the context is a complex yet believable reality like the one that is presented in *The Voyage of the Mimi*. In this multi-media science learning package, which includes videotapes, books, computer-based lab equipment, and software, students participate vicariously in the seagoing adventures of a group of scientists. The curriculum provides them with opportunities to navigate, find and release a trapped whale, and learn to solve a host of problems. Laboratory equipment allows them to study temperature, light, and sound. The *Mimi's* adventures have been carefully designed to capitalize on the known interests of students in the upper elementary school grades and to involve them in real mathematics and science contexts. As one overview of special-needs students' learning with these materials points out: "While the materials provide a very rich environment of ideas, it is nonetheless a contained universe, with more reiteration than the everyday world. Rather than overwhelming the child with academic problems, it seems to facilitate connections between concepts, activities, meanings, and outcomes" (Martin, 1986).

Other simulations are based on fantasy. These fantasy situations capitalize on children's ability to immerse themselves in a small world, identify with fictional characters, and become involved in the drama of the story. The motivational contexts of these sometimes elaborate simulations are designed to get students involved in problem solving (see Chapter 4). What could be more compelling than weaving one's way through a web of mystery and intrigue in order to find out who committed the crime as in *Where in the World is Carmen Sandiego?* or *Snooper Troops*.

Good motivational contexts can be far simpler than those described above. For example, *The Pond* provides a limited but compelling context of a frog traveling from one side of the pond to another through a pattern of leaps on lily pads. In this software, the simple frog and lily pad context is *integral* to solving a problem about pattern and sequence. We have seen students very motivated by the task of moving the frog to land only on lily pads—not in the water! Similarly, the simple context of filling a truck with 8 gallons of liquid, when you only have 6-gallon and 7-gallon containers from which to fill it (*Puzzle Tanks*), provides a compelling context for numerical reasoning (see Chapter 2).

On the other hand, not all fantasy or simulated real-world contexts work to motivate thinking and engagement. Some contexts are tangential to the learning task. For example, several programs are illustrated throughout with royal characters who introduce games, give directions, and distribute rewards. Unlike the frog-and-lily-pad or truck-and-containers, the royalty context has nothing to do with the learning goals of the software. While it may offer some initial attraction, its motivational appeal is limited.

Other contexts appear to simulate real-world situations, but fail to actively engage students in thinking carefully about problems because of serious flaws in their simulated realities. Here are two flaws to watch for:

1. An elaborate simulated context is established, but students are allowed no choices in how they solve the problems presented. Several popular mathematics simulations purport to engage students in running a store, constructing a house, or carrying out other complex activities which require planning and estimation as

well as calculation. However, what the programs actually do is to require students to carry out certain mathematical problems in a predetermined sequence. In fact, these problems are not simulations at all. Like the word problems at the end of the chapter in the math book, they provide further computation practice, but do not help students apply their skills to "real-life" situations.

2. The context allows random choices to lead to the desired results. This problem is the opposite of the no-choice situation described above. Here, students do make a range of choices, but the choices are not connected clearly to consequences. Since a random or poorly-thought-out strategy is likely to succeed, the motivation to make careful, reasoned choices is lost. For example, the many software simulations in which children try to make money by selling some commodity have been criticized because there is no limit to the number of trials students have. One way to accumulate money quickly is to find a marginally successful strategy for earning money (this is not difficult) and then to mindlessly play as many games in succession as possible before your computer time is used up. There is little motivation to think through a better strategy and, therefore, to begin to consider the economic ideas which the game is meant to teach (Lepper, 1985).

At the other end of the spectrum from simulations, whether real-world or fantasy, are computer applications which are, in fact, real ones. Using the computer as a tool for communication, creation, calculation, scientific experimentation, or storage and manipulation of information, that is, in ways that most closely parallel the actual uses of the computer in society, may be the most compelling contexts for students. Students recognize that databases, spreadsheets, telecommunications, word processors, graphing tools, programming languages, tools that collect scientific data, and so forth, are adult tools that give them real knowledge and power in the world. If it is difficult to imagine your students using such advanced computer tools, consider the following examples:

1. At a school for learning disabled children, students study heat, temperature, and sound by collecting and graphing their own data using microcomputer-based laboratory probes connected to a microcomputer (Zuman & Weinberg, 1988).
2. Special-needs students who wrote essays and sent them via telecommunications to a group of students in another part of the world produced much better writing samples than students who wrote similar essays for their own teachers (Cohen & Riel, 1986).
3. A high school special education class used databases to store information from their research on post-school jobs and electronic

spreadsheets to construct sample budgets for living on what they would earn from the jobs they had selected (M. Erandmeyer, personal communication).

In each instance, students produced work superior to what might have been expected of them. The real-world computer tools provided contexts in which these students could carry out real-world tasks—scientific investigation, communication, research, and mathematical reasoning.

A final word about context and intrinsic motivation: The teacher is an active participant in setting the context. While certain software features may tend to enhance or undermine intrinsic motivation, the teacher is key in creating or choosing intrinsically motivating contexts and matching them appropriately to students. In many of the examples above and throughout this book, it was only the teacher's timely and appropriate interventions that made it possible for the student-computer interaction to work. We will turn to a further examination of the teacher's role in the next chapter, but now we take a look at two more categories of software features that can affect learning to learn.

Enough Guidance, Enough Choice: Software Features That Promote Students' Interaction with the Learning Task

What is it about certain pieces of software that encourages children to explore, to use different strategies, even to talk aloud to themselves as they are attempting to solve a problem? Most often, this software *demands a high degree of meaningful interaction with a task that students find interesting*. Software has various features which encourage interaction, and many of these not only get students involved but help them become better learners as well. Below, we discuss features of software which:

1. Allow students to make choices.
2. Provide alternative routes into the task.
3. Set limits on time, turns, and difficulty level.

One issue that should be considered in selecting software that optimizes interaction is the *degree to which the student is allowed to make choices*. With some software, students have a wide choice in selecting goals and ways to reach the goal. Story-making software, for example, gives students freedom to write and illustrate stories within the constraints of the software's capabilities. Some of this software also gives students a range of pre-set choices, including a built-in set of drawings of common objects, or stories that can be changed or continued.

Other software allows choice of strategies for reaching a pre-set goal. Interaction with the learning task can be encouraged through the judicious use of prompts about what to do next. Particularly for students who are used to structured learning situations, on-screen prompts make it possible and safe to try alternatives to solving a problem. Software like *Where in the World is Carmen Sandiego?* helps a student organize an approach to learning by listing a small number of options that can be tried at each decision point (e.g., look at the airline schedule, visit the museum, use the on-screen computer to identify the suspect). Limiting the realm of possibilities allows students who are new to problem-solving environments to make meaningful choices.

When using software that gives students a larger number of choices, often the teacher needs to provide some structure for the situations. Students who find it difficult to make choices need support from the teacher in order to avoid feeling overwhelmed by a situation in which there are too many options. One way to help students learn to choose is to provide them at first with a small set of choices which can gradually be expanded. For example, teachers begin to teach Logo with a very small set of commands and/or some built-in shapes which students can use, thus providing appropriate constraints within which students still have flexibility and choice. With experience, children will gradually feel more comfortable in making choices within their learning tasks.

However, much of the software used with special-needs students offers *too few*, rather than too many choices. When confronted with software that prestructures every step of the student's work, little deliberation is needed. Sometimes success can be achieved simply through trial and error, which means that students are reinforced for employing no particular learning strategy. Even worse, some programs lead students too rigidly through successive steps so that success at the task seems to have nothing much to do with their own effort. *Software that provides little choice also provides little in the way of learning to be a learner.*

Related to the appropriate level of choice is the existence of *alternative routes* into the task. Not only does each student make his or her own choices, but one student can make a *different set of choices* from what another student makes, yet be equally successful. A student is much more likely to get involved in the learning task when presented with different kinds of information, encouraged to use different channels for getting information, and enabled to reach the goal by using many different learning strategies. *The Mystery at Pine Crest Manor* (a story from the *Tales of Mystery* program) provides an excellent example of an effective combination of these features. The mystery provides many different kinds of clues (visual clues, information from files, information about where things happened) and allows students to explore these clues in any manner they choose. Based on information collected from different sources, students solve the mystery by following their own path of clues. Different routes may have different advantages: for example, some strategies for solving the problem may be safe, but slow, while others may be quicker, but less reliable. Students who are methodical and those who are more impulsive can develop their own successful approaches and, with the guidance of an adult, compare the strengths of these alternatives.

Software that uses multiple routes for presenting information and for solving problems has been characterized as "thought-provoking" software, because it encourages students to move back and forth between different ways of thinking about problems (Dickson, 1985). Learners, particularly those with special needs, should be encouraged to use all kinds of thinking—including logical, verbal, mathematical, and graphical. It is only through practice with all of these modalities that students will be able to approach a new learning problem in a flexible and effective manner. By using software that has multiple routes to a goal, students learn about the advantages and disadvantages of different strategies; they learn that no one approach is right for all situations; they become aware of their own particular strengths and learn to value the complementary strengths of others, especially if appreciation of these individual differences is promoted in the classroom.

Finally, a high level of interaction with the learning task may be accomplished by setting time limits or by offering successive levels of difficulty. Such constraints (or incentives) can increase the level of challenge and encourage students to reach the goal as quickly or efficiently as possible. These features, however, can also have negative effects on students' learning strategies.

On the positive side, some simulations use the passage of time to increase the sense of reality and drama in the adventure. For example, in *Agent U.S.A.*, students must pay attention to time (as well as to many other factors) in order to make their train connections on schedule. Such limitations may help the impulsive child by encouraging him or her to make *better* choices rather than simply trying everything in a random manner. However, passage of time which the user cannot control at all may be inhibiting, even frightening. A command which allows the student to "stop time," freezing the action on the screen, is an important feature of *Agent U.S.A.* In *Where in the World is Carmen Sandiego?*, a limited amount of time is allotted to solve the mystery (a week passes in simulated time), but the user has unlimited time at each decision point to make the next choice.

On the other hand, some time limits are artificial and relate more to motor skills than to any realistic time demands. Many pieces of software impose time limits for finding keys and moving around the game board. There is no integral reason for this type of pressure, and, in fact, it may discourage students from making meaningful choices. Arbitrary time limits encourage a "video arcade" mentality: Students begin to feel that the only goal is to score as many points as possible in a given amount of time.

Many good pieces of software impose no time limitations or give students the option for timed or untimed activity. Without limitations, students feel free to explore different strategies, focusing on the problem itself, rather than on the time limit. *Unless there is a specific reason to put the additional pressure on a student, it is usually better to select software that allows the student as much time as is needed for exploration or allows students to choose their own time limitations.*

Like time limits, increasing levels of difficulty are designed to challenge students to improve their performance. When software has multiple levels of difficulty, it is easier to use it in a classroom setting with students who have a variety of levels of skills and self-confidence. When students are free to move on to a harder level *at their own pace,* it makes it easier for them to get a sense of mastery over the problem. Software that automatically moves a child on to harder problems when they have barely mastered a simpler set of problems can have negative effects on self-esteem. Children may feel that they are never able to succeed, that just when they feel close to mastering the material, they start making mistakes again (Lepper, 1985). This never-quite-good-enough effect could certainly contribute to feelings of "learned helplessness." However, when children can control the difficulty level, they are free to decide when they are ready for new challenges. Teachers can then help students find a level of "manageable complexity" (Hull, 1970), encouraging them to stick with a task for long enough and to challenge themselves to take reasonable risks.

Software Features That Tell Students How They Are Doing: Feedback and Rewards

Of crucial importance in learning to be a learner is developing the ability to interpret and use feedback in subsequent problem solving. Learner-centered software, by definition, provides feedback that is informational and that guides the learner in making successive approximations to the problem (see Chapter 1). *Neutral feedback,* rather than praise, is the best way to encourage children to take responsibility for their own learning.

In order for students to learn how to use information to improve their work, feedback must be *available* to the learner. Unfortunately, some software penalizes learners for getting help. What message does this convey to students about the learning process? It probably reinforces the notion that asking for help means that you are not very smart. In reality, most learning involves seeking help from a variety of sources. When adults perform a new task on their jobs, for example, they may seek the advice of others who have performed that task, consult the files for information on how it's been done before, and perhaps find additional references in a library. All of this help seeking is seen as legitimate, competent, and conscientious problem-solving behavior. We want to encourage the same attitude among our students: learning must involve the incorporation of *new* information obtained from whatever sources are appropriate.

Feedback comes in a variety of forms. It can be a clear presentation of the student's work-in-progress, a comparison of the student's work to a desired or predicted outcome, or an additional piece of information which will help students take the next step. Hints or clues are often used to help students progress. If a variety of clues is provided, students with different approaches can use the type of information which is most helpful to them. For example, *Mystery Sentences*, which presents cloze exercises to students (see Chapter 2), offers clues to the meaning of the sentence and also allows students to see individual letters to help them better predict what certain words might be. Hints may also be keyed to difficulty level. *Moptown Hotel* not only gives a different hint each time the learner enters an incorrect answer, but it also gives more explicit hints at the simpler levels of the game.

Such a series of different kinds of information allows students to build a richer understanding of the task gradually. We learn by slowly accumulating and assimilating ideas about something new. When a learner is stuck, he or she finds it frustrating to see a single clue repeated endlessly. The software should provide more than a single chance to get help and more than a single way out of a dilemma!

What if, despite whatever help is available, we make mistakes along the way? To return to the context of starting out in a new job, it is hard to imagine beginning such a venture without making some mistakes. In the real world, mistakes may lead to information which helps improve our understanding and performance, but our errors can also have a range of other consequences. Again, good software mimics reality in that it provides consequences that are *natural* rather than artificial. For example, in *The Pond*, a wrong move means the frog hops in the water rather than onto a lily pad. In more complex simulations (e.g., adventures like *Oregon Trail*), mistakes lead to setbacks that can be overcome by using a better strategy the next time around. Sometimes, as in real life, you can "get by" with mistakes but sometimes you cannot.

What happens when we successfully solve a real problem? Usually, the consequence is a sense of closure that comes from solving a problem, finding a pattern, or finishing a task. There is a certain amount of satisfaction to be gained simply by stepping back and seeing the final product (such as a thoroughly weeded, healthy garden) or in knowing that the task was well done. Psychologists have found that this knowledge of and satisfaction in successful completion is a powerful source of reinforcement for children (Stevenson, 1970). In study after study, knowing about the *results* of their learning has been found to be more important to children than getting *rewards* for right answers.

Thus, it is not surprising that students enjoy adventure and mystery games in which the only reward for reaching the goal or solving the mystery is the intrinsic sense of closure, completion, and mastery. Anyone who has seen a child's proud expression when she finishes writing a story with a word processing program knows how powerful the sense of completion and mastery can be. At the same time, it is perfectly legitimate for the software to include some small reward or recognition of a child's achievement (for example, having Gertrude the duck fly in with a treasure in *Gertrude's Puzzles*), as long as the *primary* reward is the sense of mastery which comes from performing the task well. This sense of closure is often missing from software where the goal is simply to rack up points, avoid making too many errors, or stave off disaster for as long as possible.

GUIDELINES FOR SELECTING SOFTWARE

In selecting software, we need both to look at the intellectual content and to determine whether the software will help the child learn to be a better learner. The following questions, summarizing the software characteristics reviewed in the previous sections, provide guidance in selecting software that promotes intrinsic motivation, offers appropriate levels of choice, and supports self-monitoring and perseverance.

- How does the software draw the child in? Is it primarily through "bells and whistles" or through tapping the child's curiosity about the subject matter?
- What is the appeal of this context to children? Does it build on existing interests, create the desire to learn about new areas?

- How is the context of the software related to the learning goal? Are the two integrally linked, or is the context peripheral to the learning goal?
- Do the motivational features distract children from the task at hand, or help them pay attention to relevant dimensions of the task?
- Does the software impose time limits? Are these limits reasonable or unnecessary for this type of activity? Does the way they are used imply that speed and automatic responses are more important than thinking?
- Does the software encourage the user to make choices? Are there ways to help students gradually make better and better choices? Do prompts about what to do next discourage the user from making decisions, or do they provide appropriate structure and support for decision making?
- Does the software provide multiple routes for getting information and [illegible]g problems? Can different students use different a[illegible] successfully in using this software?
- C[illegible]s engage in the software at different levels of difficulty or complexity? How is the level of difficulty controlled?
- Is help available to the user? Do hints or feedback provide information that helps the students get closer and closer to a solution? Is there a penalty for getting help? If several hints are available, do they simply repeat the same sort of help or do they build on each other?
- How is success or failure evaluated? Are the consequences of success and failure natural rather than artificial? Is completion or closure an important criterion of success?

HELPING STUDENTS LEARN TO BE LEARNERS

Even if software is well designed to give students appropriate messages about learning to learn, they may need help in hearing these messages. Those students who have a helpless orientation to learning need support to begin to see their successes as their own and their failures as subject to change and improvement. Students who begin to work successfully with the computer need help to generalize this learning to new arenas. For example, students who have learned that it takes many revisions to write a story using the word processor or many trials to test a hypothesis when using problem-solving software may not use this understanding in other situations. They may see work at the computer as separate from other schoolwork and be unaware that their experiences there are applicable to off-computer tasks.

In helping students to make best use of these learning-to-learn experiences, teachers can clearly establish learning-to-learn goals and make these goals explicit for the student as well as for the teacher. While what fits for each child varies widely, a number of learning-to-learn objectives are frequently mentioned by teachers. They note that their students need to learn to:

- Establish a goal
- Establish criteria for success
- Recognize what helps them get started
- Use errors as information
- Tolerate a period of confusion or frustration without giving up
- Choose among several options
- Try a new approach when stuck
- Pay particular attention to visual (auditory, etc.) cues
- Ask for help appropriately (for some children, this may mean less frequently; for others, more frequently)
- Evaluate their own progress

Working on learning-to-learn goals requires as much planning, intervention, and assessment as for other curriculum goals. For example, if a child is learning to *establish appropriate goals*, perhaps in writing, and *evaluate her own progress* towards those goals, the teacher helps her gradually take over more and more of the evaluation process, perhaps grouping her with peers who can provide collaboration. If a child is learning to *pay attention to visual information*, because that is one of the ways he learns best, he needs to be reminded to locate—or even create—visual cues. If a child is learning to *tolerate a period of frustration without giving up*, the teacher can remind her about situations in which she used mistakes as information: "Remember when you solved the *Snooper Troops* mystery? Remember how long it took and how you and Sally got so frustrated the first time you worked on it? But then, by the second or third time, you realized you were getting somewhere. I think this math lesson is like that—it feels really hard at first, but you need to work at it for two or three more sessions, and then you'll begin to understand it."

Learning is uncomfortable. While it can be satisfying and even exhilarating, for most of us the times of satisfaction are balanced by times of frustration and confusion. As adults, we know these times well. From tasks that appear straightforward (rearranging the furniture, planning a meal for guests, making vacation plans) to major new undertakings (learning a foreign language, raising a child, deciding how to present your own ideas to an audience), we have all been through periods when we did not think we would ever solve a particular problem, when we felt frustrated, angry, puzzled, helpless, and overwhelmed. Think of a learning experience you have had as an adult. It probably was not smooth sailing, but you may have known enough about learning and your own ability to learn that you could imagine that you would eventually muddle through and achieve the understanding you were seeking.

When we are in the midst of difficulty, whether in school or in life, it is often hard to remember that we *did* figure out ways to manage other situations which seemed, at the time, to be equally problematic. Getting students to remember and make use of their previous experience in learning how to learn is one of the key teaching tasks. Learner-centered software can provide appropriate environments for learning how to learn, but it is the teacher's work with the student that enables special-needs students to make the best use of this learning. In the next chapter, we explore more fully the teacher's role.

NOTES

Teachers who want to delve into real-world applications of the computer as a tool have an increasing number of software opportunities. *AppleWorks* provides an integrated package of word processor, database, and spreadsheet—"integrated" because data from one file can easily be transferred and incorporated into the others. For example, a budget prepared on the spreadsheet can be integrated into a word-processed report. *Bank Street School Filer* and *PFS:File/Report* are examples of databases in which students can enter their own data as well as use curriculum databases (in social studies, science, etc.) to organize and interpret information.

Microcomputer-based laboratories—using the computer as a tool in the science lab—are available in programs such as *Lab Experiments in Motion; Sound;* and *Heat and Temperature* (three separate packages) and *Science Toolkit* modules.

Many teachers are exploring the motivational and instructional aspects of telecommunications programs in the classroom and the context they provide for students to communicate with their peers in other parts of the country and world. One such program is the National Geographic Kids Network, a telecommunications-based elementary science curriculum.

REFERENCES

Adelman, H., & Taylor, L. (1977). Two steps toward improving learning for students with (and without) "learning problems." *Journal of Learning Disabilities, 10,* 455–461.

Cohen, M., & Riel, M. M. (1986). *Computer networking: Creating real audiences for students' writing*. Technical Report #15. Interactive Technology Laboratory, University of California, San Diego.
Dickson, W. P. (1985). Thought-provoking software: Juxtaposing symbol systems. *Educational Researcher, 14*(5), 30–38.
Hull, W. P. (1970). Attribute games and thinking skills. *The ESS reader*. Newton, MA: Elementary Science Study.
Lepper, M. R. (1985). Microcomputers in education: Motivational and social issues. *American Psychologist, 40*(1), 1–18.
Licht, B. G., & Dweck, C. S. (1984). Determinants of academic achievement: The interaction of children's achievement orientations with skill area. *Developmental Psychology, 20*(4), 628–636.
Malone, T. W. (1981). Toward a theory of intrinsically motivating instruction. *Cognitive Science, 4*, 333–369.
Martin, L. W. W. (1986). Enriching science and math instruction with technology: *The Voyage of the Mimi. Churchill Forum, 9*, 1, 2–3.
Meltzer, L. (1984). Cognitive assessment and the diagnosis of learning problems. In M. D. Levine & P. Satz (Eds.), *Middle childhood: Development and dysfunction* (pp. 131–152). Baltimore: University Park Press.
National Geographic Kids Network. Elementary science curriculum under development by Technical Education Research Centers and the National Geographic Society. Funded by the National Science Foundation, Grant #MDR8652120, and the National Geographic Society. For more information, contact the National Geographic Society, Washington, DC.
Paris, S. G., Lipson, M. Y., and Wixson, K. K. (1983). Becoming a strategic reader. *Contemporary Education Psychology, 8*, 293–316.
Pearl, R. (1982). LD children's attributions for success and failure: A replication with a labeled LD sample. *Learning Disabilities Quarterly, 5*, 173–176.
Pearl, R., Bryan, T. H., & Donahue, M. (1980). Learning-disabled children's attributions for success and failure. *Learning Disabilities Quarterly, 3*, 3–9.
Stevenson, H. W. (1970). Learning in children. In P. H. Mussen (Ed.), *Carmichael's manual of child psychology* (pp. 849–938). New York: Wiley.
Zuman, J., & Weinberg, A. S. (1988). Special-needs students: Learning to design experiments with MBL. *Hands On!, 11*(1). Cambridge, MA: Technical Education Research Centers.

ADDITIONAL RESOURCE

Licht, B.G (October 1983). Cognitive-motivational factors that contribute to the achievement of learning-disabled children. *Journal of Learning Disabilities, 1*[illegible] 483-490.

6. The Teacher's Role

We are constantly having arguments about software. Time after time, one educator begins to explain why she would *never* use a particular piece of software with her students, when another responds, "But that's been the perfect tool for helping Julie begin to have more positive interactions with her peers," or "But it's just right for my first grade students with language delays." These interactions point to a thread which has wound its way throughout the previous chapters and which we now focus on more deliberately. This thread is the teacher's role.

If the computer is truly an educational tool, then we can never look only at the *student-computer* interaction in an educational setting. Unlike the student-television interaction, which is usually unmediated, the educational use of computers necessarily has *three* components: student, computer, and teacher. The computer can be the medium of learning and teaching, but it is the student and teacher (and possibly others as well) who are weaving, through this medium, the complex relationships of education. Even if the teacher is not physically present while the student uses the computer, she is constantly making choices grounded in what she knows about the large number of uniquely individual minds with which she is actively engaged. These choices include what software to use with whom, how to allocate computer time, what expectations to have for students, when to intervene, when to refrain from intervening, which students to group together, and so forth. And these minute-by-minute judgments—based as they are both on knowledge about what options are available and knowledge about particular students—must vary widely from student to student, from teacher to teacher, from setting to setting.

In this way, using the computer is no different, and certainly no easier, than using any other educational material. But while most of the roles teachers play in the student-computer-teacher interaction are not unique to this aspect of teaching, certain roles appear to become more prominent in this setting. In this chapter, we discuss the roles teachers have found critical as they incorporated computer use into instruction. These roles include: introducer, technical advisor, arranger, visitor, silent partner, booster, mentor, and learner. The next sections of the chapter define each of these roles and describe teachers' experiences in carrying them out.

INTRODUCER: GETTING STUDENTS STARTED

Most teachers *briefly* introduce a new piece of software in some formal way, either to the whole class or to a small group which will then teach other groups. This introduction is usually designed to familiarize students with special keys and commands, screen formats, the goal of the activity, ways of making choices, and so forth. The introduction is short and participatory. Teachers suggest minimizing talky introductions. Rather, help kids get the tools they need to get started: talking about alternative strategies and ways of proceeding can happen more productively later on after students have some initial experience. Students don't have to know everything about the software in order to start. For example, students can begin using a word processor with a very small subset of commands (see Chapter 3). Programs which are gamelike or center on solving a problem (see Chapters 2 and 4) do not have to be fully explained. Students are much more willing to try something to see what happens while they are becoming familiar with a program than are most adults. If the activity is motivating, students will often ignore or overcome difficulties which we adults think will prevent them from using the program. As one resource room teacher described her experience: "I kept running over like a mother hen. I thought they wouldn't be able to handle it, but *they* were fine."

Those students who are reluctant to fail or to show their ignorance may need specific encouragement to try: "You know, when I use a new computer program, and I can't figure out what to do next, sometimes I just try a few things and see what happens." By the way, one other thing students will ignore is directions, so don't count on the directions or tutorial to help students understand what is going on. A poster made by you, or by a knowledgeable student, of important keys or commands, is probably more useful than most on-screen instructions or accompanying manuals.

TECHNICAL ADVISOR: HELPING STUDENTS OVER MECHANICAL HURDLES

Some students will have trouble mastering the *mechanics* of particular programs. Learning when to use particular keys, how to move from one section of the program to another, or how to move around the screen may require practice before students can become involved in the problem or activity. Some teachers provide time for students to practice particular mechanics—such as getting to the train on time in *Agent U.S.A.*—before they begin to work on the problem. Like separating handwriting practice from composing a piece of writing, this separation allows students to overcome mechanical barriers during a time when they are not also involved in thinking hard about their ideas.

When students' problems with mechanics persist beyond what can be helped by one or two brief practice sessions, it is not worth spending more time on these mechanical barriers, which are, after all, only means to get to the program's content. While the knowledgeable teacher wants to help students overcome hurdles appropriately, so that these hurdles do not become barriers, she does not want practice on the hurdles to become a barrier in itself. Even keyboarding skills, while important to students' eventual fluency on the computer, can be practiced *concurrently* with exposure to real and engaging uses of the computer. If keyboarding is a hurdle which students must overcome before they can do anything interesting, they are likely to conclude that the computer is just another boring vehicle for rote learning.

For some students with mechanical difficulties, programs may be selected that use a very small number of keys or which can be operated with a mouse or joystick. A number of excellent programs require a great deal of thinking and decision making, but use only two or three keys: *The Factory* requires only the spacebar and left and right arrow keys; much of what students do in the *Explore-a-Story* series is completely controlled by the mouse. Organizing information is another mechanical hurdle for some students. Many teachers devise supports to help their students record information. Teachers may provide, for example, a map outline, a list of characters with space to record critical information, or a worksheet on which students can record what they have tried and what the results have been.

Both practice sessions for software mechanics and support devices to help students organize information emerge in response to particular student needs. Sensitive teachers do not automatically require all students to spend a certain number of practice sessions on the mechanics of a piece of software nor do they assume that the path to a particular piece of software must be paved with worksheets. Judging *who* needs *what* kind of help *when* is the critical job of the teacher as Technical Advisor.

ARRANGER: BUILDING PEER COLLABORATION AND COOPERATION

A few students may find the mechanical hurdles of some software insurmountable, yet they are eager and capable in thinking about the content of the software. Although highly motivated, these students may have so much difficulty with selecting particular keys or remembering commands that they are unable to make much progress. Some students have extreme trouble finding keys on the keyboard, no matter how few keys are required or how often they have used the program before. In these instances, students can be paired with students who have more mechanical facility; often the same students who cannot find keys on the keyboard can provide excellent insights into solving the problem at hand. As one student said in protest when a teacher suggested he wasn't letting his partner, a student who was experiencing such mechanical difficulties, use the keyboard: "But he's doing all the *thinking!*"

Arranging peer collaboration is an important role of the teacher for many reasons beyond helping students with mechanical difficulties. There is growing, consistent evidence that cooperative peer work promotes learning *and* enhances social growth and development. The fact that computers have been a scarce resource may have led educators to make more use of peer collaboration in this context, but while this approach may have begun as a matter of necessity and convenience, teachers have consistently found that it has led to instructional benefits.

Many teachers have found problem-solving software (see Chapter 4) to be the perfect vehicle for developing peer collaboration and cooperation. Groups of two or three appear to work best; teachers often help the groups define and rotate particular roles. For instance, in using *Where in the World is Carmen Sandiego?*, one student might keep a list of the suspect's characteristics (red hair, drives a convertible, likes mountain climbing) while one student uses the keyboard and the third student uses the atlas to trace geography clues. In some of the more elaborate simulations that take many sessions to solve, such as *Snooper Troops*, the amount of recording, notetaking, and mapping needed to solve the problem requires cooperative work. Students learn that in order to be successful they must be willing to depend on each other and must be responsible in carrying out their individual tasks.

Teachers who use word processing with their students make extensive use of peer collaboration (see Chapter 3). Writing is a cyclical process in which composing is followed by a period of stepping back and considering what has been written. It is during this reflective stage that peers can act as audience and critic. There is also a diagnostic/evaluative advantage to having students work in pairs or threes. When students work together, they need to talk to each other, to decide what to do next, and sometimes to argue for their own ideas. Teachers can learn much more about how their students are thinking by listening to such dialogues than by watching a student working alone silently.

VISITOR: CHECKING IN AND HANGING AROUND

Most teachers are unable to sit with their students while they work at the computer, and, in any case, such close supervision may not be desirable. While students move toward greater self-confidence and autonomy, they may need to take some risks in private. But many teachers circulate, stopping in to see what students are doing, asking a question, suggesting an approach, or just giving verbal support. At the beginning, more frequent checking in may be required. Students who are not used to making choices, taking risks, or engaging with problems which take a long time to solve may need gradual weaning from teacher support. As Steve Spencer, the teacher of a self-contained learning disabilities class put it: "I hung around a lot as they were playing, tossing in ideas, suggesting strategies, and helping if someone became stymied" (see Chapter 4). Planning for the students who are not working on the computer is an important part of managing computer use. In the beginning, teachers often schedule computer time for some students and independent seatwork for others so that this "hanging around" is possible.

Keep in mind that the need to hang around will decrease as students become more familiar and proficient with a particular piece of software and more able to help each other. One of the important characteristics of a teacher as a Visitor is that he does not appear too often and he leaves before his welcome wears out.

SILENT PARTNER: FOSTERING STUDENT INDEPENDENCE

While some of the teacher roles are active and participatory, the essence of this important role is *refraining from action*. This role may be one of the hardest teachers have to play. Our impulse as teachers is to help. Especially when we are working with students who do not have confidence in their own thinking, we try to prevent them from experiencing too much failure. As Donna Simone, a high school resource teacher put it, "We tend to be there for them: *We're anticipating what they're not going to do before they don't do it.*"

Yet one of the common learning-to-learn goals (see Chapter 5) that teachers have for their students with special needs is increased independence—a combination of having confidence in and taking responsibility for their own work. The computer tends to support student independence. If students are using software that allows them to try something and see the results of what they tried *without waiting for the teacher*, they may find themselves in a position they have experienced only rarely—being able to evaluate and change their own work. One teacher, commenting on the contrast she saw in her students' work on and off the computer, observed, "If you're stuck with paper and pencil, you're just stuck. With the computer, you can usually try something."

Many teachers have noticed that their students begin to take steps toward greater independence when they use the computer. The problem for teachers is to remain on the sidelines long enough to let students take these steps. Recognizing and supporting this fledgling independence requires careful observation and some overcoming of our protective impulses. Donna Simone talks about her experience this way:

> In knowing your individual students, you learn to anticipate—you're aware of when they're going to need help or at what point they may break down. You end up kind of keeping an eye on their work and pacing them in your mind, so you know at what point you're going to get there a second before they do distract the others. And what's happening with the computer is—they're not needing me! I caught myself wanting to go before they needed me! One of my students said, "not yet, I want to figure it out myself." *You* want to figure something out yourself? I find that I still have to watch myself and pull myself back and wait until clearly they need me. And what I'm trying to do now is even make them wait a little bit longer and push them a little bit the other way. It takes a lot of work to do that, though. I still find myself hovering near certain people.

As this teacher expresses so well, it is difficult to shove the young birds out of the nest, but that push is crucial for their ultimate survival.

BOOSTER: DEALING WITH STUDENT FRUSTRATION

One of the consequences of refraining from always being available to help is that students will experience times of confusion and frustration. We want students to learn to cope with such periods, which are an inevitable part of the learning process, but we also realize that students will not become totally independent learners overnight. Everyone needs a boost sometimes.

Teachers have found it helpful to set up explicit procedures for what one teacher calls "frustration times." Peer collaboration groups can sometimes address this need. Other teachers have a schedule of individual conference times (for talking about a piece of writing, for example) or identify student experts responsible for helping with particular pieces of software. Others meet with small working groups or gather the whole class to help students to describe, clarify, plan, hypothesize, and predict.

Group discussions can focus on such questions as: What happened in your last session at the computer? Did you get stuck? Why do you think that might have happened? Did anyone else get stuck at the same place? What will you try next? Why do you think that might work? When using some computer software, such as problem-solving software, it is often true that no one has been completely successful, so that it is not a matter of "winners" telling "losers" what to do; everyone is on an equal footing, sharing partial success—much more like real life!

In situations in which one group of students is using the computer while others are engaged in other work, the teacher is often hard-pressed to give needed help. Many teachers make use of student experts to assist with questions they don't have time to answer. Students learn that when they are stuck, don't understand directions, or have forgotten which keys to use, they can call on Sue. Sue, of course, has permission to leave her own work temporarily when she is called on for help. Not only does this arrangement make the teacher's job easier, but use of student experts demonstrates to students that expertise does not reside only in the teacher.

MENTOR: REQUIRING STUDENT REFLECTION

All the roles we have discussed so far help the student connect with the computer as an enabling tool that may help her or him learn, but these roles only set the stage. At the core of the teacher's role is, after all, teaching. And in an interactive environment, where the student is engaged in computer use, the teacher is the Mentor who nudges, irritates, questions, challenges, listens, and in every way attempts to keep the students active as learners and thinkers rather than passive responders.

This role leads the teacher to be *demanding* of his or her students. It is not enough for them to spend their time at the computer quietly or to finish an activity. The questions are: What did they *do* with their time? What were they thinking about? What difficulties did they encounter? What are they pleased about?

In essence, the Mentor requires children to be reflective about their own work, to evaluate, question, and be critical of what they have accomplished. Mary Briggs, a resource room teacher who had been taking some of her fourth graders to the computer lab to use *Logo*, thought of her mentoring role as one in which she helped these learning disabled students think hard about each of their projects:

> I think we've worked very hard to be rigorously analytic. Each child has been given the space to speak and to listen and to be talked to in looking at exactly why they've made the choices they've made. I don't just let them do something and say, "fine." Every single thing we look at and say, "well, what about this and what about that and why did you do that?" Everyone has the freedom and the safety to say what they want to say. And I guess that is what I wish somebody had done with me. I think this approach is what they can then take with them when they come into new problem settings.

Reflection is also the key for Betty Church, a resource room teacher who uses the word processor as a means to free students to think about what they want to say in their writing: "I think basically what I'm trying to do is have the kids reflect on their own experiences, their own feelings and thoughts, and hopefully in the process of writing about these they begin to grow and expand as people."

LEARNER: TAKING RISKS IN PUBLIC

While the computer does not replace the teaching role of the teacher, it often makes more visible a role which students do not see often enough: teacher as Learner. Teachers who become involved with the computer typically find more and more potential computer uses they would like to explore, but the time available does not expand to accommodate this interest. The struggle to find enough time to learn about new software while trying to keep up with the daily demands of students, curriculum planning, and paperwork can be frustrating. Because there is never enough time to learn and plan, teachers must often learn along with their students. As it turns out, this arrangement has its advantages, as Betty Church explains:

> The children are beginning to see me as a real human being. And I think that's great, because kids somehow have this idea that their teachers are these marvelous, well-put-together people who can do anything. And I think what's happening with all of this is that they're seeing me as a person who has similar kinds of strengths and weaknesses. When they saw me working on this word-processing program, they saw that it didn't just happen, that I really had to work to learn how to do it.

Many teachers have found that learning about the computer leads to an expansion of roles for both student and teacher: The teacher can be a learner, and the student can be a teacher. Sometimes students might even learn something that the teacher doesn't know yet! In a high-school special education English class, most of the students and the teacher used *Bank Street Writer*, but students were encouraged to learn about other software which was available in the room. Donna Simone recounts:

> We have two kids who learned *AppleWorks*, and they've become so totally independent, it's like a marvel for a special educator. I can't get at their work because I don't know how to get at it because I don't know *AppleWorks* well enough, so I can't even go over and help them. They've got to figure it out themselves. One of them taught himself the Speller that goes with it and then taught the other one, and I haven't learned it yet.

While taking the risk of trying something you haven't mastered in front of your students may be a bit frightening, putting oneself in the role of Learner has its rewards as well. Not only do students learn that every learner encounters difficulties and experiences frustration, but also teachers find that opening up to these risks and frustrations ultimately provides renewal for their own teaching. Donna says:

> I've been teaching special ed for 8 years, and it was wonderful how starting to use the computer made me rethink what I do. I had gotten to thinking, "well, I know all this stuff works real well and it works for me, and if it works for me, I'm going to pretty much stick with it." Well, there's all this other great stuff that you discover when you begin to rethink how you present things and come up with other options.

Betty Church feels the computer has opened her up again to possibilities and ideals she had stopped considering. When asked about the change, she laughed and said, "I feel like a first year teacher. I think it's great!"

While there are certainly other teacher roles which arise in the course of using the computer with students, the eight roles discussed in this chapter capture what many computer-using special educators see as the essence of their work: getting students started, helping students over mechanical hurdles, building peer collaboration and cooperation, checking in and hanging around, fostering student independence, dealing with student frustration, requiring student reflection, and taking risks in public. They provide a core of teaching processes which are critical in moving beyond drill and practice, both in fact and in spirit.

RESOURCES

For further reading on teacher roles and the teaching profession, we recommend:

Duckworth, E. (1987). Teaching as research. In E. Duckworth, *"The having of wonderful ideas" and other essays on teaching and learning* (pp. 122–140). New York: Teachers College Press.

Hawkins, D. (1974). I, thou, and it. In D. Hawkins, *The informed vision: Essays on learning and human nature* (pp 48–62). New York: Agathon Press.

Okazawa-Rey, M., Anderson. J., & Trover, R. (1987). *Teachers, teaching, and teacher education*. Cambridge, MA: Harvard Educational Review.

7. Getting Started

By now we hope that using the computer with your special-needs students sounds intriguing and possible. You are ready to find some software that works for you and to try this learner-centered software with your students. Much as we wish we could give you a recipe for success, you can see that using the computer is like any teaching tool. You must decide how and when to introduce it, and make decisions about which children will use it under what circumstances. In order to help you take the beginning steps, this chapter is structured to focus on the first implementation issues. We try to help you decide how to tell what software is good; how to start out with the students; and how to find out about software. Finally, we offer names and addresses of software companies, magazines, and other resources.

HOW CAN I TELL WHAT IS GOOD?

When you find a piece of software which challenges and interests you, the chances are that it will have the same effect on some of your students. You will probably become interested in those pieces of software which are learner-centered (see Chapter 1). Think about why a particular piece of software appeals to you. Is it interesting? Motivating? Why? Does it sustain interest over time? Is it challenging? Does it match your students' interests?

You know your own preferences. Your own learning style and your preferences deeply affect your teaching style. Some teachers prefer to teach students how to do procedures step-by-step; others prefer to let students generate their own solution procedures. Some teachers prefer to break down tasks for their students; others would rather have students wrestle with complexity. What you prefer in your teaching has a great deal to do with your own preferences in learning.

The more you learn about your own learning style, the more you will understand the students who are similar to you. At the same time, however, remember that some of your students learn in different ways than you. They will like many pieces of software which you do not find appealing initially; they will want to try out activities which you find boring. They may not want to read directions when you always do; they may not want to try new puzzles when you always do. In any case, remember that as you become more and more expert in finding your own learning preferences with the computer, you can also discover more and more about your students' preferences and how they may differ from yours. The computer provides a good arena for talking about style and learning about style.

Stay alert to your own learning style, as it limits your choices, too! Teachers who use a variety of software with their special-needs students find, over and over, that their predictive capabilities are sometimes wrong. Students with severe problems with directionality are often drawn to *The Factory*, which requires interest in orientation and spatial visualization. Students who seem unable to make decisions may spend time working on *Botanical Gardens*, solving problems which require complex decision making. Young learning disabled students are able to work extremely effectively with *Logo*, which requires planning, risk taking, and curiosity. *Talking Text Writer* has convinced many teachers that learning disabled students can learn to write fluently as they learn to read, using the word processor to write stories that are read back to them by the computer. Their writing quickly moves beyond the over-simplified story form of the usually unenthusiastic writer.

If you decide a piece of software is too complex, go ahead and give it to some students to preview! Someone will probably love it and master it. Taking the role of researcher and observer will help you to watch your students as they use software in the classroom. Collect data consciously—ask students questions; listen to your students as they work together. You will learn a great deal about your students' interests and learning styles this way. Moreover, you will learn your own "blind spots" and will be more aware of your own preferences as well as your students' preferences as you make decisions about use and purchase of software.

Getting started can be easy or hard—it depends on you, your sense of humor, and the appropriateness of your choices for your students. If you can take a deep breath, select some decently promising programs, and plunge in, you are likely to enjoy it.

Trying out new pieces of software can provide you and your students with a shared exploration of something new. David Hawkins (1974) writes about a triangular "I-Thou-It" relationship among teacher, students, and materials. He suggests that these elements must be balanced in a good curriculum so that teachers' interests, students' interests, and the content are complementary. The computer can provide an "It" for both teacher and student. Focusing on something other than another skills worksheet may help you and your students develop different, more collegial, relationships. The opportunity to be a researcher *with* their teacher may create a real change in students' perception of you and your shared tasks and experiences.

Criteria for Evaluating Software

Our definition of learner-centered software (see Chapter 1) can help you ask some important questions about a new piece of software:

- Does the software support students in making meaningful choices?
- Does the software encourage or allow students to use prediction and successive approximation?
- Does the software provide information feedback to the student?
- Does the software rely on external rewards or does it engage the student in a meaningful task which can lead to intrinsic motivation?

Beyond this first level of questions, teachers find that there are many questions which help them focus on features of the software that support meaningful learning. We include a list of such questions below. In addition, you will find that you have some strong preferences about software as you search for programs you can use well over time. It is important that you and others you work with can develop some ways of looking at, screening, and recommending for or against the purchase of software. Sometimes you may be asked to order or recommend for purchase without preview. This is generally not a good idea unless you have used the software in some other context and can feel confident that it will work in your setting, with your students, supporting your goals.

Most departments develop some format for software evaluation and often order software on 30-day preview from vendors. During this preview period teachers try the software with selected students and look carefully at it, thoughtfully weighing its advantages and disadvantages. In that way, when the budget process begins, they are prepared with a list of priorities that they feel confident will enhance their collections.

In order to evaluate software for your own uses, you will probably want to develop a more formal checklist of elements to consider. Many teachers use some combination of the following, but this list is definitely not exhaustive. Rather, it is suggestive of some of the categories you may want to attend to. There are software evaluation sources which are mentioned in the resource section of the book as well, and many of the educational software magazines regularly review new pieces. We suggest you review the list of questions in Chapter 5 (Learning to be a Learner) and think about the list in the following software checklist as well.

Characteristics of the software

- Does it require any special hardware or peripheral equipment (e.g., a second disk drive, enhanced memory, a mouse)?

- Is it visually appealing? Are the graphics clean and clear?
- Are icons or symbols used or is a particular reading level expected?
- Is there a teacher management system?

User control

- Can you ignore the directions if you want?
- Is help available while the program is running?
- Can you "escape" from the program at any time?
- What kind of feedback is given to the student?
- Can you or the student select a difficulty level?
- Can a game be saved halfway through its play?
- Must students remember complex keystroke combinations?

Content of the software

- What content does this software support?
- What themes or topics does the software deal with?
- What curriculum and curriculum extensions does it support?
- Does it sustain student interest over time, or is it "one-shot"?

Potential uses

- What learning styles do my students have?
- Does the software lend itself to collaborative learning approaches?
- What off-computer curriculum activities would be compatible with this software?
- How could I introduce this software—through large-group demonstration, through small groups, pairs, or individuals?
- Do I enjoy using this piece of software? Why or why not?

Appeal to students

- What is most appealing to me about this software?
- What learning styles does it appeal to?
- What is most appealing to three students who try this program?
- Is this interesting to work with? Enjoyable? Satisfying?
- Does the program do something worthwhile?
- Will students find this babyish? Too hard?

Flexibility

- Can I create tailor-made activities for my students?
- Can I enter some of the vocabulary or content to the program?
- Can students put their own work into the program?

- Will the program support or allow students to follow their own interests?

[Please note: this is not meant to be an extensive set of software criteria. If you want more extensive criteria, you may be interested in evaluation forms in sources such as *T.E.S.S. (The Educational Software Selector)* or in Mary Male's book *Special Magic.*]

HOW DO I START WITH MY STUDENTS?

Pick One or Two Pieces to Start with and Develop These Thoroughly. Over and over, teachers urge their colleagues not to move rapidly through many pieces of software, but to take their time. Two major reasons seem to emerge for this. First, and most important, teachers cannot simultaneously plan lessons and continue their ongoing paperwork while throwing themselves totally into a new technology. To make sensible decisions about available time and effort often means that moving slowly is the only solution. Knowing many pieces of software vaguely will not, in the long run, empower you to make good decisions about using them. Knowing a few in depth will allow you to adapt them and explore their use with a variety of students.

A second reason for not rushing through software is more symbolic. If we feel that students can zip through software with relatively little commitment to talking and thinking about what they are doing, we contribute to an entertainment mode of learning. Students will not become invested in going beyond the provided information and will become passive learners, waiting for the scene to change. They will not reach out enthusiastically to grasp new ideas, nor will they realize that they can shape their own learning. Rather, by contributing to an entertainment model we will be contributing to their passivity. Intellectual curiosity takes time to develop.

Feel confident, then, in pacing yourself as slowly as you need to. Find ways of using the computer for your own work as well. As you master the word processor or the database or the spreadsheet you will find ways to integrate it into your own curriculum. Your use of these tools for your own work will model skilled adult use of the computers—and for most of your students this will be a powerful influence indeed.

Starting with Skill Development

Some teachers start by finding one program that develops learning skills they especially value. For instance, you might select *Mystery Sentences* in order to develop reading comprehension. It can be used to develop vocabulary and reading comprehension as well as problem-solving skills. If you prefer to begin with another content area, you might start with *Power Drill* in order to practice mathematics computation in a different format. Yet another piece might be *Bumble Games*, which focuses middle-grade students on graphing and displaying number pairs. Any of these programs will help students develop particular skills, and will reinforce these skills after you have taught them. (See Chapter 2 for more details.)

Starting with Word Processing

Another way to start is to use a word processor. There are many that have features useful to special-needs students—large type, clean screens, spell-checkers, and voice synthesizers all help students to focus on their writing. Some teachers start students writing with "sentence starters"—specific beginnings which are open-ended. For some children, completing only two sentences and printing them out represents a major breakthrough, and will inspire more work.

Other teachers start using the word processor to create a "final copy" of children's written assignments for them, entering the stories and essays after school or during their free periods. Gradually, they hand over more and more of this responsibility to their students, and the students learn to enter their own work into the computer.

The voice synthesizer increases student autonomy dramatically. With a word processor such as *Talking TextWriter* the students can be independent even before they can write many sentences, entering their work and listening to the computer as it spells and sounds out their entries. Even though it reads in a "robot voice," this word processor has been shown to produce gains in students' ability to write *and* to read, beginning quite early in school (Rosegrant, 1985). Students can write their own stories and others can "read" them using the computer. This comes close to replicating the early reading experiences of many children whose parents read aloud to them—and they can easily swap stories and follow along as the computer reads. (See Chapter 3 for more details about word processing.)

Starting with Problem Solving

Problem solving is yet a third entry point. Linda Ware's work with *Moptown Parade* and Steve Voiles' work with *Gertrude's Secrets* are good models of developing materials which enhance problem solving with the computer software. Students do not always develop solution strategies naturally; [illegible] they need to have the problem presented and re-presented [illegible] ntexts.

Working with a specific piece of problem-solving software can develop into a very long curriculum. You can develop off-computer games emphasizing likeness and differences with *Moptown*-type characters and then extend them into other populations as well. Students may enjoy sorting and classifying activities which deal with flowers, birds, baseball players, professional musicians—whatever populations most pique their interests.

Further, in working with problem-solving software, connections with the language arts are an integral part of the process. You can use problem-solving formats to explore sounds, spelling, and meanings. For instance, if you introduce a classification game based on finding similarities, and the words chosen are *Sunday, wonderful, dungeon,* and *funny,* some of your students will take pleasure in the fact that they all have a short U sound. Working in these game formats with reading and writing vocabulary has a motivating effect on many students. (See Chapter 4 for more ideas.)

INTEGRATING AND EXTENDING SOFTWARE USE

You will probably find that the first two or three pieces of software which you decide to use with the students will become the backbone of your computer program for the first year or two. Adding to that repertoire will become important, but moving slowly into new areas is something which most experienced computer-using teachers recommend.

Dawna Traversi, a moderate-special-needs teacher, reinforces the idea of moving slowly:

> I really felt that the amount of time spent on the computer doing word processing was the most valuable. When you do something and you find it's working, you find yourself expanding on that rather than seeking out something in addition to that.

At first the computer may be relatively isolated from your ongoing curriculum, but you will find more ways to integrate it as you go along.

Developing a "Researcher" Mindset

When you start something new in the classroom you have a unique opportunity to take a researcher's position about it and find out what works well with different students and why. Introducing a new piece of software or a new learning unit based on a program can give you the chance to do some observation and to collect data. Once teachers enter the classroom after student teaching, there are few opportunities for observation sustained over time. This introduction of a new technology allows you to take the "action researcher" position.

First, you may not be certain how to proceed. You may want to be completely honest with your students, telling them that you are not sure what they will learn from this software that you do not already teach them, so you will be watching and talking with them and taking notes. Sometimes this is enough to make the students interested in the process.

Second, you may find it hard to take notes and to interview your students without directly intervening in their learning process. It is hard to discipline yourself to *listen* and to clarify students' comments without trying to coach or correct. If you can simply follow their process as they work, you will be able to learn much more. Sometimes writing down an entire conversation as two students work at the computer will give you a great deal of insight into their process with a particular program, and into their understandings and misunderstandings.

Third, you will find that it is a good idea to look at your notes and data for insight rather than for evaluation. It is not only whether a piece of software "works" or not—much more important to you is *how* it is affecting your students' thinking. Are they able to engage in the kinds of strategy which the program requires? Can you develop off-computer activities which will support them by presenting skills similar to those used in the software? (See Linda Ware's work in Chapter 5.) How does the computer support students' cooperative learning? What other activities could you do to support similar strategies?

A helpful mental stance is that you really *do not* know how something will work, or in what ways it will attract your students. Keeping track as it unfolds (over years, in some cases) will allow you to develop your hunches into theories, and your theories into action plans. With such a collection of data, you will be able to explain and interpret computer use in your classroom and also be ready to order new programs whenever budgets open up.

The hardest thing for many teachers is tolerating the confusion they feel when their students enjoy working with software and are clearly improving their skills, but curriculum extensions based on the software are not yet quite fully refined. Pressure to do traditional work, where the outcome is carefully specified, can be hard to overcome. Students may make comments about how much fun this stage is—and how unlike their "regular" work. You may well become a little concerned, wondering whether the "fun" is precluding learning; this concern is natural but unfounded. Clearly, however, the "middle muddle" can be difficult, because neither you nor the students are sure what you are up to!

During this time, you may find that you have many good ideas, and that your ideas are getting increasingly complicated and comprehensive. Some of your ideas will work; others will not pan out. You may find that by simply letting the students work with the software they will come up with new ideas themselves. Allow yourself to keep going during this period of uncertainty, and you will find that there are many rewards! Not only will your students be interested in their learning, but you will find that you can make the computer central to some pieces of curriculum, and can begin to use only a few pieces of software for many, many kinds of learning.

Finding and Using Time

Learning to use the computer well is a long, slow process. It may take you a whole year to develop the use of one good piece of software, and you may spend 2 or 3 years before you feel thoroughly knowledgeable in using it with your classes. This time is well invested, but many teachers seem to assume that they should quickly master the details and nuances of using the computer with their students. Be sure that you spend adequate time; if you content yourself with using many, many pieces of software superficially, you will not learn much about what works and how to build on your successes.

When you use the computer for your own paperwork, be prepared to invest time again! One of the enduring myths is that you will save time immediately using the computer. That *is* true in the long run, but in the short run you have to make a significant investment of time. Learning to use the machine, to troubleshoot when things go wrong, and to evaluate software carefully will take a good deal of time.

Many experienced special-needs teachers laugh as they realize how much time they spent in mastering the new medium and in planning for their students' learning. Most of them are well aware that they were not told quite how consuming the work could become; on the other hand, the rewards are there in proportion to the amount of time invested in the planning and exploration. For many teachers the excitement of finding something new and motivating to use with their students is enough to keep them going for a long period of time, and the rewards of using the materials to simplify their own work are substantial. Imagine—updating announcements to parents, using the computer to keep track of students' progress, doing unit planning on a machine which allows easy copying, editing, and revision! Many teachers are enthusiastic about using the machine for their own school work, whether their students use it or not!

It is much more important to use a few pieces of software in depth than to dabble in many. When you master one piece you can integrate it intelligently into your own curriculum, building on the lessons you have tried and your knowledge of your own students' needs.

Likewise, it is better that the students learn to use one piece well than to scatter themselves through a variety of programs, leaving them whenever they are tired of them. When you find the right pieces of software you will be able to build lots of learning activities on them.

Special-needs teachers' time is a very valuable commodity, and it is not necessary to use the computer for many of the learning activities which are traditional in the resource room. Rather than replan everything to fit onto the computer or to fit into existing software constraints, it is important to think about the "unique uses" of the computer—in other words, using the computer for things which *it* can offer rather than using it as a pale imitation of other learning materials. The computer is much more than an animated worksheet or a way to entertain two students while you work with another eight. It is important that you find ways to match its unique uses with your students' unique needs and your unique creativity. A spreadsheet is much more than an elaborate adding machine; a word processor offers much more than a typewriter; a database is much more useful than a group of file cards.

Helping yourself and your students find the power and satisfaction of real learning is truly an exciting endeavor. The computer in and of itself will not do that, but your use of it may provide some opportunities to extend your curriculum and include some new and exciting challenges in handling complex information and wrestling with real-world problems and applications. Good luck on the journey!

HOW CAN I FIND OUT ABOUT SOFTWARE?

Through People

The computer provides some uniquely frustrating experiences for all teachers. Not only are there unknowns to face about how your students will react, but there are the problems that arise whenever you rely on machines. Some teachers find these problems very frustrating; others take them with a grain of salt.

Every computer-using teacher has had to change plans at a moment's notice when he or she found that the disk counted on for today's lesson was not compatible with the available machine, or when he or she found that the only available printer was out for repairs. Using the computer is not about perfection; any technologically based teaching is subject to small but annoying glitches. This is what makes support systems necessary and extremely helpful. You are not alone in your uncertainty! The question is not whether you will need a support system, but how to develop one.

One major source of information will be your colleagues. If someone in your school system is using computers with special-needs students and you are curious about it, take the time to talk with that person. You might visit the classroom as well, so that you can see it in action. Find out what's working, what kinds of management techniques she uses. Does she introduce new software to the whole group and have pairs follow up with it during class time? Does she do off-computer activities which introduce, integrate, and reinforce the computer work?

Dawna Traversi recommends visiting as a way to get started:

> I think that it's important to visit other situations where other teachers are using the computer. I've learned a lot from seeing other teachers use the computer, having kids explain to me what they're doing, how they're thinking about it, what they're excited about, what parts they like best. I think just becoming exposed to situations where computers are in use is one good thing. You don't need to sit there reading a manual, but you really should see it in action. And then once you've done that then decide what is most useful to you.

If no one in your system is using computers with special-needs students, perhaps a teacher in a nearby system is. You might make arrangements for a professional visiting day so that you can observe his or her computer teaching. It's often worth taking the time in order to see and talk with someone who's very experienced, or who is having the same concerns you are.

Your "regular education" colleagues who use the computer have a great deal to recommend and to share as well. You might find the most active computer user in your school and interview or visit him. Chances are he has developed methods of using the computer which will help you make some decisions. Certainly he will know how you can get access to the software which is available in your system; he will have recommendations about the most useful starting points.

Students may represent a source of help for you. Often a student who uses the computer at home or who has extensive experience in another context will be happy to be a resource for you. You can develop a sense of worth in many students by training them to do the troubleshooting for the entire class. They can all learn a checklist to consult *before* asking you for help (check the plug, check the switch, see if you've booted the software, etc.).

Your system's computer coordinator may be delighted to provide you with some starting points. Not only do most coordinators want to help teachers, but some take on the challenge of special-needs students with real zest. Your coordinator may also be able to help with software and with other resources. He or she may be able to provide you with some information about conferences, publications, and upcoming inservice workshops which might be of use to you. If not, you will find that there are many people to turn to with your questions.

In one system, Tom Plati, the computer coordinator, began by challenging the special-needs teachers in the high school to try using word processing with their students. As they started to do so, he was available for telephone consultations and for troubleshooting. Tom spent time in the classroom working with the teachers as they were beginning the instructional process. He learned a great deal about the special-needs students; the special-needs teachers learned about the computer and its use in the writing process. Much to the teachers' joy, many of the students mastered word processing and went on to use it confidently and well. Tom, of course, was pleased to see that his hunches had been borne out. Since the beginning, these professionals have expanded their work with the computer to include a great deal of other software, including spreadsheets and graphing software in the math classes. From this initial work, collaborative work evolved. The computer coordinator worked with the special education teachers to develop a grant for purchasing computers and software specifically for the special-needs students in the system.

Support groups, or Special Interest Groups (SIGs), can often be formed for the benefit of new computer users as well as the more experienced. At Technical Education Research Centers (TERC) we formed the Microcomputers in Special Education/Special Interest Group (microSIG), a network of special-needs teachers in the greater Boston area. It provides an opportunity for participants to talk about their own work in the classroom and their own experiences, preview software with other teachers, borrow software to try in their own classrooms, and attend presentations on new equipment, software, and computer applications. These elements could form the nucleus of a special interest group in any location.

Teachers welcome the opportunity to learn from their colleagues. In many systems the special-needs departments are isolated from other teachers and do not feel involved in curriculum change. Special-interest groups can give participants a chance to learn about changes as they occur; a software library gives participants a chance to try out software with their own students rather than to purchase it without knowing whether it meets curricular goals. Most important, however, time needs to be provided for teachers to talk with each other about their work in the classroom.

If there are a number of teachers who want information, you may want to consider beginning a special-interest group. There may be enough interest in your school district or in a number of districts in your area. Continuing monthly or bimonthly meetings will provide a framework and a forum for interested colleagues, and you may well find some business or organization in your area which is willing to sponsor or partially sponsor the costs of mailings and software acquisition. Whether or not you can cover a large geographic area, you will find like-minded colleagues and begin the kind of professional dialogue which will enrich everyone's teaching.

You may want to develop a number of different sources of support as you begin to use the computer more extensively—at some times more technological and at other times more emotional. Generally, you will find that other people will be interested in your experiences and will want to share their own with you!

Through Books and Magazines

Every year there are more publications that address special-needs students' computer uses. Some publications focus on computer uses, as *Closing the Gap* does—it contains a wealth of information. Check with your department head if you don't subscribe to these yourself, or ask your librarian to help you track down some copies. Often you will find that there is a person in the school system who subscribes to them, and you can ask that they be routed to you as well. Many publications contain extremely helpful essays written by computer-using special educators that recommend specifics on how to use programs and how to organize the children.

It's important not to overlook the "regular education" sources for information on computers in the classroom. We include a list of some of those publications at the end of this chapter. Special-needs teachers' questions about how to use, how to set up, and how to extend the use of pieces of software are not always very different from those of the regular classroom teacher. If you learn well from a written source, this could be one of the most helpful references for you. Your colleagues and the school librarian should be able to be of help here.

Further, your computer coordinator may be able to help you find some of the most relevant computer-oriented publications. Often the magazines which are written for the broadest computer use include many articles oriented toward large business users. Some, however, are oriented to small individual users. The magazines published for users of the machines you have in school may be helpful to you—and chances are that your computer department will subscribe to them. We have included a list of resources at the end of the chapter.

Through Workshops and Courses

Your school system's inservice coordinator may be helpful in scheduling some inservice training sessions for the special-needs teachers in your system. Often the people who plan inservice training welcome input from interested teachers. There may well be some budget money put aside which could be used to hire a course facilitator who uses computers with special-needs students.

The most effective inservice courses extend for more than one session and include some follow-up work in classrooms. If you can, try to plan one which includes those elements. When you are trying to learn a great deal of new information and are planning to integrate it into your teaching, you need as much hands-on experience as possible.

You may be able to negotiate with the school to borrow a computer to use at home while you are learning about software. More and more schools are making computers available to teachers during the summer and school vacation. It is very helpful to have this access, so that you can work directly with the computer and can try out a variety of techniques and pieces of software when you have the time to explore it in some depth. Often your household insurance will cover the computer in this situation. If not, you may want to purchase a relatively inexpensive (perhaps used) computer for your own work.

Charlotte Reid, a high school special-needs teacher, says:

> Now I really think that the computer in the teacher's home as well as a computer in the classroom is a real priority. And I find I carry a box of disks back and forth rather than stacks of worksheets and my life has changed a lot.

You may find that your school system has some computers which are specifically designated for participants in training workshops; if not, there's certainly no harm in asking! Parent-teacher groups sometimes donate money for teachers' professional equipment. This might be one route to try.

Dawna Traversi recommends not limiting yourself to workshops geared exclusively for special-needs teachers:

> I think that workshops could be more of a collaboration with regular ed teachers as well as special ed teachers because it's just an eclectic approach—the sped teacher can pick out what will work with your kids and you just disregard the stuff that just doesn't seem to fit. And so if you're hearing how regular teachers are using the computers it's just helpful to you in a different kind of way. You can make the modifications and adaptations that you need to make and you get really excited about it when you see that somebody else is doing something. They just click something different in your mind and then you just take off. So the more time spent with *any* other teacher or student using the computer is useful.

Charlotte Reid suggests you start with a course in word processing at a university or a computer center.

> That's what my students use all the time. They do their poetry and their essays and their letters on the computer and I use it to do all of my paperwork too. Starting with word processing gives you the most powerful tool right at the beginning.

Through Conferences

In many professional organizations, the regional conference includes sessions on using the computer with students. Generally there are one or two sessions which are specifically geared to using the computer with special-needs students. At times you may find that the "regular" teachers' conferences are helpful to you—there may be sessions introducing or demonstrating software or sessions which deal with managing the use of the computer in the classroom. Check out the schedule when one of your colleagues has received it. You may well make important contacts at these events.

Conference planners, like inservice coordinators, often welcome ideas for sessions. You may find that you can ask someone in your regional association whether they are routinely including special-needs and computer sessions in the conference offerings. If they are not, this is an opportunity to suggest that they do. There may be demand for sessions which will help "regular" teachers deal with special-needs students through appropriate use of computer software. Sessions in collaborative or cooperative learning, in using tool software, in creating databases which will help you keep your own records, can all be shared among teachers with a wide variety of specialties.

The more you request sessions on learner-centered software and the more you assume that others are interested in the creative uses of the computer with special-needs students, the more you will find that planners will be happy to work with you to design conference offerings that go beyond drill and practice and begin to address the more challenging and fruitful uses of the computer in the special-needs classroom.

Through Previewing Centers

You may want to preview software in a course, a university computer center near you, or in a resource center in your school system. Some software dealers or manufacturers allow a 30-day preview of software. Your coordinator may know which ones do. Try software yourself and *with your students.* Find one or two pieces of software which you really like and work with them until you feel confident that you want to use them with some of your students. Many teachers have started by using a word-processing package with their students; others have started with software which is of more limited use, such as *Moptown Parade* or *Where in the World is Carmen Sandiego?* If you prefer group introductions to new material, work out your scheme and begin!

REFERENCES

Hawkins, D. (1974). I, thou, and it. In D. Hawkins, *The informed vision: Essays on learning and human nature* (pp. 48–62). New York: Agathon Press.

Rosegrant, T. (1985). Using the microcomputer as a tool for learning to read and write. *Journal of Learning Disabilities 18*, 113–115.

DIRECTORY OF RESOURCES

The purpose of this section is to provide a selection of resources that will help you take the next step—and keep going!

This section is divided into two parts. The first lists all the software identified in the text, noting publisher, computer(s) supported and chapter in which it was described. To help you locate the software, a compilation of publishers' addresses follows the software listing. In the second part, we offer a selection of print and organizational references.

Software

The software described in this book represents a selective sample: "selective" because we and our teacher-contributors have used these programs successfully with some students; "sample" because the programs do not represent all the high-quality software available. Omission of a learner-centered software program indicates that we have not used it regularly or we simply did not choose it to illustrate points made in the book. We offer the software sampler in Table 1 as a starting place for beginners and as new ideas for experienced teachers.

Since software costs vary depending on vendor, quantity, and packaging (lab pack, school edition, etc.), we have not listed prices. Most single programs mentioned here cost from $40 to $80. Software tools, such as word processors, *Logo*, spreadsheets, and microcomputer-based labs, are usually more expensive and may cost from $100 to $300. However, their versatility and expansiveness often justify the cost.

Every program listed here is available for the Apple computer, usually for several models, and we have also noted availability for other computers. You should, however, check with the software publisher because more programs are becoming available for other hardware, especially IBM.

TABLE 1
Software Sampler

Title (Publisher)	*Computers Supported*[a]	*Chapter Number*[b]
Agent U.S.A. (Scholastic Software)	A,C,MS,AT	4
AppleWorks (Claris Corp.)	A	2
Bank Street School Filer (Sunburst Communications)	A,C	5
Bank Street Speller (Scholastic Software)	A,C	3
Bank Street Story Book (Mindscape Inc.)	A,C,I,T	2
Bank Street Writer (Scholastic Software)	A,C	3
Bank Street Writer III (Scholastic Software)	A,T,I	3
Botanical Gardens (Sunburst Communications)	A	7
Bumble Games (Learning Co.)	A	2
Bumble Plot (Learning Co.)	A	2
Crossword Magic (Mindscape Inc.)	A,C,I,T,AT	2
Enchanted Forest (Sunburst Communications)	A,I,T	4
Explore-a-Story (Collamore/D.C. Heath)	A	2
Exploring Tables and Graphs (Weekly Reader Software)	A	2
Fractions (Control Data)	A	2
FrEdWriter (public domain software)	A	3
Gapper Anthology (Queue)	A	2
Gapper Reading Lab (Queue)	A	2
Gears (Sunburst Communications)	A,C,I,T,TR	4
Gertrude's Puzzles (Learning Co.)	A	4
Gertrude's Secrets (Learning Co.)	A	4
Hinky Pinky (Mindscape Inc.)	A	2

(Continued)

TABLE 1 *(Continued)*
Software Sampler

Title (Publisher)	*Computers Supported*[a]	*Chapter Number*[b]
How the West was One + Three x Four (Sunburst)	A	2
Kidwriter (Spinnaker Software)	A,C,I	2
Koala Pad (Koala Technologies, Corp.)	A,C,I	2
Lab Experiments in Motion, Sound, Heat and Temperature (Queue)	A	5
Logo (Logo Computer Systems Inc.)	A	2
Logo (Terrapin)	A,C,Ma	2
Logo (IBM)	I	2
MacDraw (Claris Corp.)	Ma	3
MacWrite (Claris Corp.)	Ma	3
Magic Slate (Sunburst Communications)	A	3
MECC Graph (MECC)	A	2
Microcomputer-based Labs (Queue)	A	5
Missing Links (Sunburst Communications)	A,C,I,TR	2
Moptown Hotel (Learning Co.)	A	5
Mystery Sentences (Scholastic Software)	A	2
My Words (Hartley)	A	3
Newsroom (Springboard)	A,C,I	2
Number Quest (Sunburst Communications)	A	2
Oregon Trail (MECC)	A	5
PFS: File/Report (Scholastic)	A,I	5
PFS: Write (Scholastic)	A,I	3
Power Drill (Sunburst Communications)	A,C,I,T,TR	2
Proteus (Research Design Associates)	A	3
Puzzle Tanks (Sunburst Communications)	A,C,I,T,TR	2
Quations (Scholastic Software)	A	2
Science Toolkit (Broderbund Software, Inc.)	A,I	5
Sensible Speller (Sensible Software, Inc.)	A	3
Snooper Troops (Spinnaker)	A,C,I	4
Storymaker (Scholastic Software)	A	2
Success with Typing (Scholastic Software)	A,I	3
Super Print (Scholastic Software)	A,I	2
Tales of Mystery (Scholastic Software)	A,C,I	5
Talking TextWriter (Scholastic Software)	A,I	3

(Continued)

TABLE 1 *(Continued)*
Software Sampler

Title (Publisher)	*Computers Supported*[a]	*Chapter Number*[b]
The Factory (Sunburst Communications)	A,C,I,T,TR	6
The Pond (Sunburst Communications)	A,C,I,T,TR	4
The Print Shop (Broderbund Software Inc.)	A,C,I,T	2
The Voyage of the Mimi (Holt, Rinehart & Winston)	A	5
TimeLiner (Tom Snyder Productions)	A	2
Type to Learn (Sunburst Communications)	A,I	3
Where in Europe is Carmen Sandiego (Broderbund)	A,I,C	4
Where in the USA is Carmen Sandiego (Broderbund)	A,I,C	4
Where in the World is Carmen Sandiego (Broderbund)	A,Ma	4,5
Wizard of Words (Advanced Ideas)	A,I,C,Ma	2
Wonderful World of Paws (South-Western Publishing Co.)	A,I,C	3
Word Quest (Sunburst Communications)	A,C	2
Word-A-Mation (Sunburst Communications)	A	2
Write Connection (Scott, Foresman)	A	2
Write On! (Humanities Software)	A,I,Ma	3

[a]A = Apple II family, C = Commodore, I = IBM and compatibles, T = Tandy, Ma = Macintosh, TR = TRS-80, AT = Atari. Please note that not all Apple II software is completely compatible with each II in the family.

[b]If a program is described in the text, the chapter is referenced.

Note: Sometimes the price of software seems prohibitive. For those who are willing to explore software of uneven quality, some of the public domain programs are relatively learner-centered. With such software it is permissible to make copies to give to students and colleagues. One excellent program is *FrEdWriter,* a powerful word processor. It and other public domain software are available through the Boston Computer Society, One Center Plaza, Boston, MA 02108. Phone (617) 367-8080. Another source is Softswap, 33 Main St., Redwood City, CA 94063. Phone (415) 363-5472.

Software Publishers

Advanced Ideas
2550 Ninth St.
Suite 104
Berkeley, CA 04710
415-526-9100

Broderbund Software Inc.
17 Paul Drive
San Rafael, CA 94903-2101
800-527-6263 or 415-492-3500

Claris Corp.
440 Clyde Avenue
Mountain View, CA 19493
415-960-1500

Collamore/D.C. Heath
125 Spring Street
Lexington, MA 02173
800-334-3284 or 617-862-6650

Control Data
P.O. Box 0
Minneapolis, MN 55440
800-828-8001 or 612-853-8096

Hartley Courseware, Inc.
Box 419
Dimondale, MI 48821
800-247-1380 or 517-646-6458

Holt, Rinehart & Winston
383 Madison Avenue
New York, NY 10017
212-750-1330

Humanities Software
P.O. Box 950
Hood River, OR 97031
509-493-1395

IBM
PC Software Department TR
One Culvert Road
Dayton, NJ 08810
800-IBM-2468 ext. 900/TK

Koala Technologies Corporation
269 Mt. Hermon Road
Scotts Valley, CA 95066
800-223-3022 or 408-438-0946

Learning Co.
6493 Kaiser Drive
Fremont, CA 94555
800-852-2255 or 415-792-2101

Logo Computer Systems, Inc.
555 W. 57th St., Suite 1236
New York, NY 10019
212-765-4780

MECC
3490 Lexington Ave. North
St. Paul, MN 55126
800-228-3504 or 612-481-3500

Mindscape Inc.
3444 Dundee Rd.
Northbrook, IL 60015
800-221-9884 or 312-480-7667

Queue
562 Boston Ave.
Bridgeport, CT 06610
800-232-2224 or 203-335-0906

Research Design Associates
P.O. Box 848
Stonybrook, NY 11790

Scholastic Software
730 Broadway
New York, NY 10003
212-505-3000

Scott, Foresman and Company
1900 East Lake Ave.
Glenview, IL 60025-9969
312-729-3000

Sensible Software, Inc.
210 S. Woodward
Suite 229
Birmingham, MI 48011
313-258-5566

South-Western Publishing Co.
5101 Madison Road
Cincinnati, OH 45227
513-271-8811

Spinnaker Software
One Kendall Square
Cambridge, MA 02139
800-826-0706 or 617-494-1200

Springboard
7808 Creekridge Circle
Minneapolis, MN 55435
612-944-3915

Sunburst Communications
39 Washington Ave.
Pleasantville, NY 10570
900-431-1934 or 914-769-5030

Terrapin Logo
376 Washington Street
Malden, MA 02148
617-492-8816

Tom Snyder Productions
90 Sherman St.
Cambridge, MA 02140
800-342-0236 or 617-876-4433

Weekly Reader Software
10 Station Place
Norfolk, CT 06058
800-327-1473 or 203-542-5553

Print and Organizational References

This book presents much information, but using computers with students is an ever-changing proposition—new software, new techniques, new issues. Of course you will continue to use what has worked well for you, but it is helpful to keep abreast of new happenings. In addition, you may have questions about specific software and hardware, certain student populations, particular types of applications, etc. Here we offer a selection of print and organizational references to assist you.

Magazines: Computers in Education. Computing education magazines providing practical information for teachers are listed below. Software reviews are a regular feature. Subscriptions are under $30.

Classroom Computer Learning
2451 East River Rd
Dayton, OH 45439
800-543-4383 or 513-294-5785

The Computing Teacher
University of Oregon
1787 Agate St.
Eugene, OR 97403
503-686-4414

Electronic Learning
730 Broadway
New York, NY 10003
212-505-3000

Teaching and Computers
730 Broadway
New York, NY 10003
212-505-3000

T.H.E. Journal
Information Synergy Inc.,
2626 S. Pullman
Santa Ana, CA 92705
714-261-0366

Magazines: General Computer. General computer magazines do not focus on education but do provide useful information on selecting, purchasing, and upgrading hardware. A multitude of such magazines exist, and many are available on local newsstands. Those listed below are a few selections.

InCider
IDG Communications
80 Elm St.
Peterborough, NH 03458

MacWorld
PCW Communications
501 Second St.
San Francisco, CA 94107

PC World
PCW Communications
501 Second St.
San Francisco, CA 94107

Reference Books: Software. If you need to review a greater number of software programs, you may want to purchase a reference such as one below. Prices vary depending on frequency issued.

The Educational Software Selector (T.E.S.S.)
EPIE Institute
P.O. Box 839
Water Mill, NY 11976
516-283-4922

Software Reviews on File
Facts on File
460 Park Ave. South
New York, NY 10016
212-683-2244

Software for Schools, 1987-1988
R. R. Bowker Co.
245 West 17th Street
New York, NY 10011
800-521-8110 or
212-645-9700

(Lists—does not review—software, cross-referenced by subject and grade level.)

Catalog: Special Education Software. To buy software especially selected for children with learning disabilities, grades K-8, request a free copy of the following catalog.

Special Times
Cambridge Development Laboratory
42 Fourth Ave.
Waltham, MA 02154
800-637-0047 or
617-890-4640

Periodicals: Computers in Special Education. The following periodicals, which focus on computer use in special education, are issued by organizations that will also provide other services as described below. Subscriptions are under $30.

Catalyst
Western Center for Microcomputers in Special Education
1259 El Camino Real
Suite 275
Menlo Park, CA 94025
415-326-6997

(Modifies, sells, and supports adaptive devices, including speech synthesizers, as well as serving as an information and referral clearinghouse.)

Closing the Gap
P.O. Box 68
Henderson, MN 56044
612-248-3294

(Offers on-site training services and an annual conference on technology in special education, as well as site licenses for a resource service which includes database and phone help-line.)

Service Organizations

Several organizations provide extensive services, which range in price from free to a few hundred dollars.

Center for Special Education Technology
1920 Association Drive
Reston, VA 22091
Hotline: 800-873-8255, 9:00 a.m.–4:30 p.m. EST, Monday-Friday

A federally-funded project that provides a range of information services to educators. Center products list local, state, and national technology resources. Guides summarize issues of technology use. Call the toll-free number with specific questions.

The Council for Exceptional Children (CEC)
1920 Association Drive
Reston, VA 22091
703-620-3660

The professional organization of special educators, CEC publishes books, reports, and bibliographies on computers in special education and sponsors related conferences and workshops. One division, Technology and Media, is devoted solely to these concerns.

IBM/Special Needs Exchange
P.O. Box 18707
Washington, DC 20036
703-439-1492

An on-line information exchange on IBM software, services, and computers, using CompuServe Information Service. Also maintains a database, accessed through an electronic bulletin board, of IBM special education software.

National Special Education Alliance
Apple Computer, Inc., Office of Special Education
20525 Mariani Ave., M/S 36-M
Cupertino, CA 95014
800-732-3131, ext. 275

A coal[illegible] of grass-roots, nonprofit organizations sponsored by Apple Computer and dedicated to bringing technology to disabled individuals by offering hands-on experience, software and hardware information, and networking. Currently there are 23 centers in 19 states and it continues to expand. Contact the Alliance for nearest center location.

SpecialNet
GTE Education Services, Inc.
National Systems Management, Inc.
2021 K St., NW, Suite 315
Washington, DC 20006
202-835-7300

An electronic service that offers subscribers a multitude of information in special education via bulletin boards and databases. Technology-related topical areas include computer applications, software, and an Apple special education solutions database. Subscribers can also communicate via electronic mail. All state departments of education and many school systems subscribe.