Claremont Colleges
Scholarship @ Claremont

CGU Faculty Publications and Research

CGU Faculty Scholarship

9-1-1988

Implementing Educational Software and Evaluating Its Academic Effectiveness: Part I

Karen Jolicoeur

Dale E. Berger
Claremont Graduate University

Recommended Citation

This article first appeared as Jolicoeur, K., & Berger, D. E. (1988). Implementing educational software and evaluating its academic effectiveness: Part I. Educational Technology, 28(9), 7-13.

Implementing Educational Software and Evaluating Its Academic Effectiveness: Part I

Karen Jolicoeur and Dale E. Berger

Two major obstacles are responsible for the delay in getting good educational software into schools. First, software implementation is a complex process that many schools are simply not prepared to undertake. Second, there is very little empirical research available on the specific factors that make educational software effective, leaving it extremely difficult to separate good from poor quality software. We will describe a basic plan for implementing educational software into classrooms, incorporating a research design that permits educational researchers to measure the effectiveness of the software. Part I of this series of two articles presents a basic plan for implementing educational software into classrooms. Part II, to appear next month, examines factors related to the academic effectiveness of eight specific software programs that were tested in several classrooms.

Common Software Implementation Problems

When teachers purchase educational software they generally buy it with the expectation that they can take it into their classroom or computer lab and students will be able to use it immediately to improve basic skills or develop new skills. Educational software comes in flashy and enticing looking packages, claiming to teach important academic skills that students will "enjoy" learning. When the software is "booted up" the vivid graphical presentations often appear quite exciting to the teacher. However, at some point many teachers discover they have no idea how to fit the software into their lesson plans. One problem is that it is not clear how most educational software relates to the standard textbook curricula, leaving teachers in a quandary about how or when to use the software. While current research does not present a method for integrating educational software with textbook curricula, it does present a method for standardizing the use of software in school settings. If teachers have software that they think is appropriate for students, the present study will provide them with a scientifically controlled plan for implementing the software, as well as a method for evaluating the academic effectiveness of the software.

Another problem teachers often face is scheduling computer time. Even though schools are acquiring microcomputers at a rapid rate, most classrooms must still share a limited number of computers for blocks of 15 to 30 minutes at a time (*Learning 86*, 1986). This can present quite a challenge if the educational software was not designed to be used in short blocks of time.

During the last several years, the main theme of the school/computer encounter has clearly shifted from "how can schools acquire computers" to "now that schools have computers, how can they best use them." In the past, controlled research has not generated much data focusing on this crucial issue. Although there is a multitude of opinions, there is little evidence currently available that supports one software implementation method over another (Becker, 1987; Jolicoeur and Berger, 1986).

Designing a Software Implementation Plan

The first step in bringing educational software into a classroom is to devise an overall plan listing the goals of the software implementation process and the major steps that will be taken to accomplish these goals. The plan developed for implementing educational software into several classrooms in the present study consisted of the following steps:

1. Specify the overall goals of the implementation procedures.
2. Select appropriate software.
3. Develop software support materials.
4. Randomly assign students to comparable groups.
5. Schedule and implement computer time for students.
6. Test student skills at regular intervals.
7. Evaluate the success of the software implementation procedures.
8. Evaluate the results of the issues examined.

Steps 4, 5, and 6 were performed by teachers in the current study. Steps 1, 2, 3, 7, and 8 were performed by the authors to standardize the evalua-

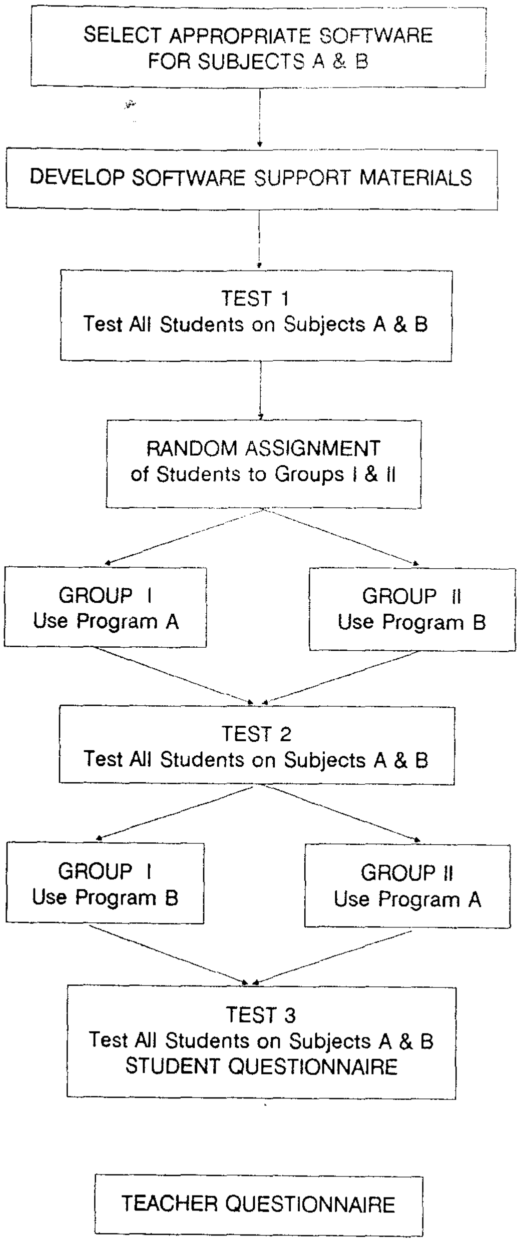
Karen Jolicoeur is an Educational Computing Consultant for Lehrer Associates, 4208 Dundee Drive, Los Angeles, California 90027. Dale E. Berger is Professor of Psychology at The Claremont Graduate School, Claremont, California. This research was supported by Apple Computer, Inc., and by Hesa, Inc., in conjunction with Project "Apple Core" that is currently in progress in the Archdiocese of Los Angeles.

tion methods and implementation procedures used in eight participating schools. However, these five steps could also be performed by teachers or administrators of a school interested in conducting their own research.

Eight fifth grade classrooms (215 students) from schools operated by The Archdiocese of Los Angeles participated in the present study. Each participating classroom was given multiple copies of two different software programs (one fraction program and one spelling program) to use for a four-week period, as well as software support materials described later in this report. All students were first tested on both subjects (Test 1) and then randomly assigned to use one of the two programs, 30-minutes a day, three days a week, for two weeks (i.e., six 30-minute sessions). Following a second test on both subjects (Test 2), the students switched programs and used the other program for the remaining two weeks. A second posttest on both subjects and a questionnaire (Test 3) were completed by all students following the four-week CAI (computer-assisted instruction) session. In addition, all participating teachers completed a questionnaire at the conclusion of the CAI session. This software implementation process is summarized in Figure 1.

Figure 1

Software Implementation Process


SELECT APPROPRIATE SOFTWARE FOR SUBJECTS A & B
DEVELOP SOFTWARE SUPPORT MATERIALS
TEST 1
Test All Students on Subjects A & B
RANDOM ASSIGNMENT
of Students to Groups I & II
GROUP I
Use Program A
GROUP II
Use Program B
TEST 2
Test All Students on Subjects A & B
GROUP I
Use Program B
GROUP II
Use Program A
TEST 3
Test All Students on Subjects A & B
STUDENT QUESTIONNAIRE
TEACHER QUESTIONNAIRE

Selecting Research Issues to Study

When students use educational software, the first question that comes to mind is: Is the software effective at teaching what it was designed to teach? If not, there is no point in continuing to use the software. It therefore becomes vital for schools to measure the effectiveness of the software programs they use.

By controlling the features of the software selected, other important issues can also be examined. In the present study, five issues were evaluated: (a) Overall software effectiveness, (b) the effectiveness of individual software programs, (c) the effectiveness of tutorials versus games, (d) knowledge retention patterns based on the type of learning processes, and (e) gender effects.

Selecting Appropriate Software

Selecting appropriate software is essential for successfully implementing educational software in a classroom and evaluating desired issues. First, the length of the lessons must match the time available for individual students. In many schools, the software must contain lessons that can be performed by students in 15 to 30 minutes (*Learning 86*, 1986). Second, the software must be easy for the students to work on their own, so they do not spend their time trying to figure out how to work the software. Finally, the software must contain material that is relevant to all issues under investigation.

Eight educational software programs that are sold commercially were used in the present study. The subject areas covered were fractions (addition and subtraction) and spelling (grade 6 spelling words). The subject of fractions was selected because there were many commercial CAI programs

Figure 2

Fraction and Spelling Software

TUTORIALS	GAMES
FRACTION	
Fractions, Decimals, Percent (SVE) *Fractions* (Eduware)	*Fraction Action* (Unicorn) *Galaxy Math: Fractions* (Random House)
SPELLING	
3 sections of *Spell It!* (Davidson) *A+ Spelling* (AEC)	Game section of *Spell It!* (Davidson) *Spellicopter* (Designware)

available from which to choose, and fractions is a concrete subject matter that most students require specific exposure to in order to learn the basic concepts. Likewise, many commercial spelling programs were also available. In addition, spelling software had the advantage of allowing the authors to use the same word lists in the selected software, thus providing tighter experimental control.

Two of the fractions and two of the spelling programs provided tutorials and/or demonstrated various methods for solving fractions problems and learning how to spell words. The other two fractions and spelling programs consisted of instructional games presented in arcade-style formats. The eight programs used in this study are listed in Figure 2.

Developing Software Support Materials

For most educational software, software support materials must be provided. While educational software is generally easy for students to use, it is not always clear to the students which portion of the program they should be working on at a particular time. This confusion frequently leads students to "bounce-around" a program haphazardly, repeating the sections they enjoy and skipping other sections entirely. Since the present study showed that students are not accurate at predicting the educational value of software (discussed in Part II), they should not be left on their own to determine the order and parts of a software program they will use. Instead, they should be led through the software in a sequential manner that matches the intentions of the software developers and the classroom's curriculum.

The **Student Instruction Card** shown in Figure 3 is one example of the software support materials developed for the present study. In all, six Student Instruction Cards were developed for each software program. These cards provided stepwise instructions for all students to follow, one card for each of the six days they used the fraction and spelling software.

Other software support materials developed for the present study included a **Teacher's Guide** designed to provide the teachers with details of the basic procedures they needed to implement the educational software into their classrooms, a **Group Assignment List** used to rank-order students according to their CTBS (Comprehensive Test of Basic Skills) scores, and a **Classroom/Computer Schedule Form** consisting of a blank calendar that the teachers completed to specify how they scheduled student computer and test times.

Assigning Students to Comparable Groups

To randomly create two comparable groups of students based on academic ability, CTBS national percentile scores for spelling and total math were summed for each student. The students in each classroom were then rank-ordered from lowest to highest CTBS sums, with even-numbered students placed in Group I (the fraction software first, then the spelling software) and the odd-numbered students placed in Group II (the spelling software first, then the fraction software).

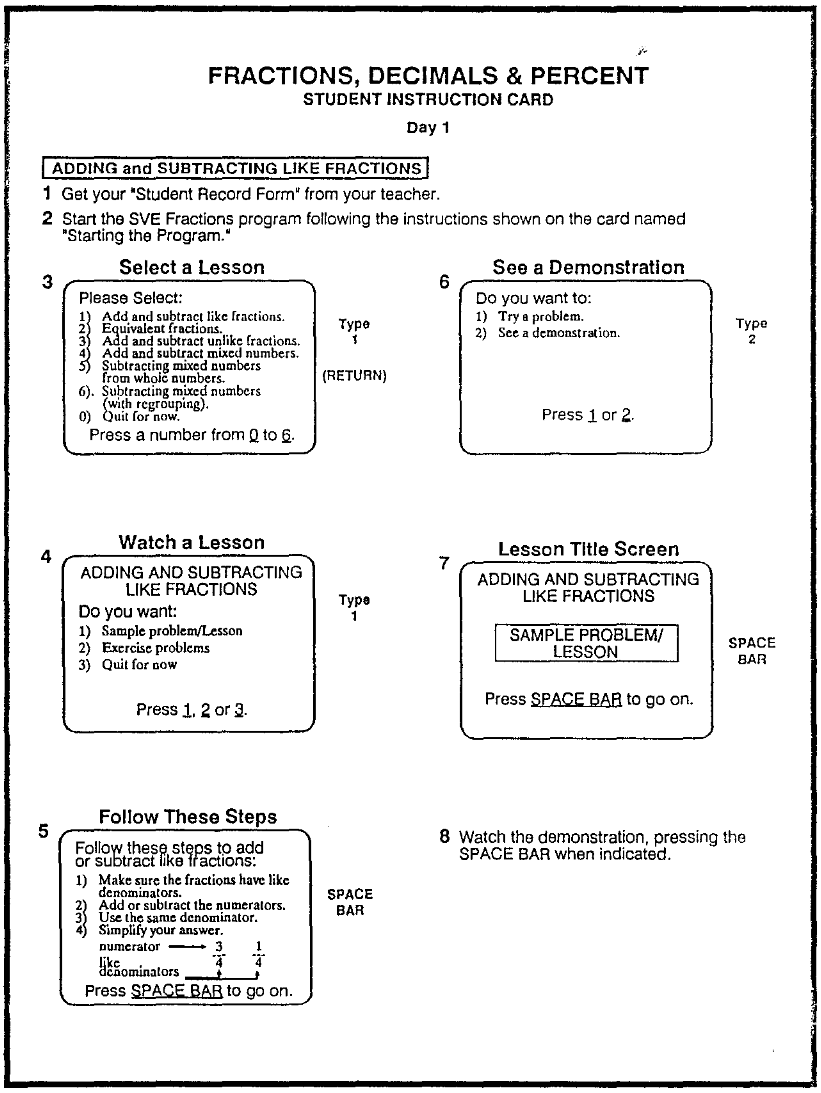
Scheduling/Implementing Computer Time

One problem that commonly arises in schools

Figure 3a

Student Instruction Card Example

Front



FRACTIONS, DECIMALS & PERCENT
STUDENT INSTRUCTION CARD
Day 1

ADDING and SUBTRACTING LIKE FRACTIONS

1 Get your "Student Record Form" from your teacher.

2 Start the SVE Fractions program following the instructions shown on the card named "Starting the Program."

3 **Select a Lesson**

Please Select:
1) Add and subtract like fractions.
2) Equivalent fractions.
3) Add and subtract unlike fractions.
4) Add and subtract mixed numbers.
5) Subtracting mixed numbers from whole numbers.
6). Subtracting mixed numbers (with regrouping).
0) Quit for now.
Press a number from 0 to 6.

Type 1 (RETURN)

4 **Watch a Lesson**

ADDING AND SUBTRACTING LIKE FRACTIONS
Do you want:
1) Sample problem/Lesson
2) Exercise problems
3) Quit for now
Press 1, 2 or 3.

Type 1

5 **Follow These Steps**

Follow these steps to add or subtract like fractions:
1) Make sure the fractions have like denominators.
2) Add or subtract the numerators.
3) Use the same denominator.
4) Simplify your answer.

numerator ⟶ $\frac{3}{4}$ $\frac{1}{4}$
like denominators

Press SPACE BAR to go on.

SPACE BAR

6 **See a Demonstration**

Do you want to:
1) Try a problem.
2) See a demonstration.
Press 1 or 2.

Type 2

7 **Lesson Title Screen**

ADDING AND SUBTRACTING LIKE FRACTIONS
SAMPLE PROBLEM/ LESSON
Press SPACE BAR to go on.

SPACE BAR

8 Watch the demonstration, pressing the SPACE BAR when indicated.

with computers is scheduling adequate computer time for the students. When teachers find software they would like their students to use, they must determine how much time is required to complete the program, and how they can split this time into smaller increments that fit their classroom's allotted computer time. To do this, several factors must be examined:

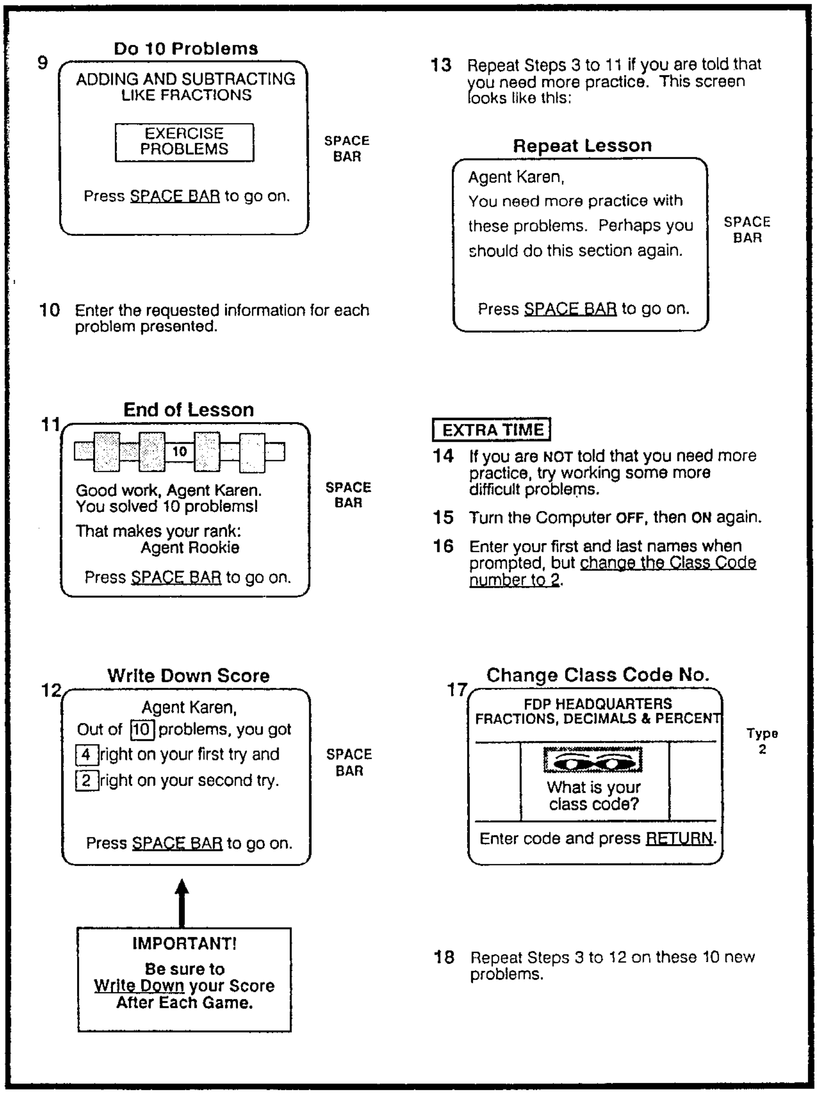
- The total number of computers available to the classroom.
- The times and frequency of computer availability.
- The number of students in the classroom.
- The student/computer ratio recommended by the software.
- The number of minutes required for each session with the software.
- The total number of sessions required to complete the software program.

Based on the above factors, the teachers in the

Figure 3b

Student Instruction Card Example

Back



9 **Do 10 Problems**

ADDING AND SUBTRACTING LIKE FRACTIONS

EXERCISE PROBLEMS

Press SPACE BAR to go on.

SPACE BAR

10 Enter the requested information for each problem presented.

11 **End of Lesson**

Good work, Agent Karen. You solved 10 problems!

That makes your rank: Agent Rookie

Press SPACE BAR to go on.

SPACE BAR

12 **Write Down Score**

Agent Karen,
Out of 10 problems, you got
4 right on your first try and
2 right on your second try.

Press SPACE BAR to go on.

SPACE BAR

IMPORTANT!
Be sure to Write Down your Score After Each Game.

13 Repeat Steps 3 to 11 if you are told that you need more practice. This screen looks like this:

Repeat Lesson

Agent Karen,
You need more practice with these problems. Perhaps you should do this section again.

Press SPACE BAR to go on.

SPACE BAR

EXTRA TIME

14 If you are NOT told that you need more practice, try working some more difficult problems.

15 Turn the Computer OFF, then ON again.

16 Enter your first and last names when prompted, but change the Class Code number to 2.

17 **Change Class Code No.**

FDP HEADQUARTERS
FRACTIONS, DECIMALS & PERCENT

What is your class code?

Enter code and press RETURN.

Type 2

18 Repeat Steps 3 to 12 on these 10 new problems.

present study planned the entire schedule for the four-week CAI sessions *before* the sessions started.

Testing Student Skills

To measure the academic effectiveness of specific educational software for individual students, students must be tested on the material presented in the software both *before* and *after* completing the software lessons. A pretest establishes baseline scores on each student which are then compared to posttest scores to determine how much each student learned during the software course.

Three versions of an achievement test measuring fractions and spelling ability were developed for the present study. The format for the questions was similar to the format used for fractions and spelling questions on the CTBS. These three tests were counterbalanced and distributed to the schools as Test 1, Test 2, and Test 3.

Evaluating the Success of the Software Implementation Plan

Software Implementation Problems. The success of the software implementation procedures was examined by reviewing the frequencies of problems encountered by the teachers implementing the software, and ratings of the student lessons by both teachers and students. Nine teachers (two from one school) who implemented the software completed the Teacher Questionnaire. On the questionnaire, they rated seven implementation procedures on a four-point scale from 1 (no problems) to 4 (many problems). The implementation procedures rated were (a) randomly assigning students to groups, (b) scheduling computer time, (c) introducing the software to the students, (d) supervising the use of two programs at one time, (e) getting students to follow the lessons, (f) getting students to record their scores, and (g) administering the three tests.

Scheduling computer time was clearly the most troublesome area with seven out of nine teachers indicating "some" or "many" problems. None of the participating schools had enough computers for all of the students in one classroom to use the computers at the same time. This meant that the students had to take turns using the computers throughout the day, thus creating numerous scheduling problems. One school had 35 fifth graders and only five computers, but managed to schedule seven students per computer per day! This, of course, consumed the entire school day on three days of the week, and meant that other students in the school could not use the computers on those days during the four-week session. However, the major complaint was that students had to miss other classes or activities in order to participate in the computer program. This created difficulties with tracking and rescheduling the fifth grade students into the events or subjects they had missed.

One way to minimize rescheduling students into missed activities would be to select CAI that students could use only once or twice per week and still benefit from the software. Then different students could be scheduled to use the computers at the same time on different days, thus minimizing the number of activities missed. For example, with 10 computers, 40 students, and software requiring use one day per week, 10 students could be scheduled to use the computers each day, Monday through Thursday, from 10:00 to 11:00 a.m. On Fridays, during the same hour, all students could work on the classroom activities they missed during their hour of computer activities earlier in the week.

Computer hardware failures also complicated scheduling computer time at two of the schools. While there isn't much that can be done to avoid unexpected equipment failures, this problem should always be considered a threat to any computer schedule. If at all possible, it is best to build flexibility into computer schedules by leaving one to two computers on reserve during each computer session. Then if equipment failures do occur, spare computers are available. It is interesting to note here that no software failures occurred on any of the 48 program disks or 24 data disks during either of two four-week computer sessions. This statement speaks for itself on the durability of floppy disks in classroom situations.

Other problems that the teachers checked were relatively minor. For instance, four teachers indicated problems randomly assigning students to groups. However, seven of eight teachers followed the random assignment procedure perfectly. The remaining teacher did not randomly assign the students to groups as instructed because she took over the classroom just prior to the computer session when the regular teacher was called away on an emergency. Instead of assigning the students to groups based on their CBTS scores, the new teacher just assigned the students to groups based on the math groups they were currently assigned to. Fortunately, this deviation had little impact on the outcome of the present study since the math groups were comparable.

Evaluation of the Student Lessons. Student Instruction Cards were designed to step the students through the software in a logical and efficient manner each day so that students could complete all sections of the software in the time allotted for each two-week computer session. Both the teachers and students completed evaluation questionnaires, rating several aspects of the lessons for both the fractions and spelling software. All of the teachers and most of the students (82 percent) agreed that the Student Instruction Cards made the computer programs easier to use.

Student Ratings of the Software. All students who participated in the present study were asked to rate several attributes of the software they used following the completion of their four-week computer sessions. Overall, the students rated the educational software favorably. On the Student Questionnaire, over 80 percent of the students said they "liked" or "loved" using the computers to learn fractions and spelling, with ratings for individual programs ranging from 2.5 to 3.5 on a 4-point scale from 1 (low rating) to 4 (high rating). We will discuss the validity of these student software ratings in Part II, based on the academic effectiveness of each of the eight educational software programs.

Summary

Part I of this series describes a successful software implementation plan for teachers who are interested in using educational software on a regular basis to supplement their classroom instruction. The prescribed software implementation plan was tested on eight fifth grade classrooms with the teachers of each classroom evaluating several aspects of the implementation process.

The most common implementation problem reported by the teachers was scheduling computer time for their students. Each teacher had a limited number of computers that students took turns using in order for the entire class to receive software instruction. This meant that students had to miss other activities when it was their turn to use the computer, creating difficulties for the teachers who were trying to track and reschedule the students into the activities they had missed. These teachers used software lessons that required each student to use a computer three days per week. One solution to minimize computer scheduling problems is for students to use educational software that they can benefit from with a limited amount of use, such as one or two days per week. This solution permits the same number of computers to be used by two to five times more students each week than software that requires use three or more days per week.

The teachers reported only minor problems with other implementation steps, such as randomly assigning students to groups, introducing the software to the students, supervising the use of two programs at one time, getting students to follow the software lessons, getting students to record their scores, and administering three tests.

In addition to measuring the success of software implementation, the software implementation plan permits educators to evaluate the academic impact of individual software programs. When the effectiveness of specific educational software is verified, computer assisted instruction can be used with confidence. In Part II, next month, we will discuss the measures and results of evaluating eight educational software programs for academic effectiveness. □

References

Becker, H.J. *The Impact of Computer Use on Children's Learning: What Research Has Shown and What It Has Not* (Report No. 18). Baltimore, MD: Johns Hopkins University, Center for Research on Elementary and Middle Schools, 1987.

Jolicoeur, K., and Berger, D.E. Do We Really Know What Makes Educational Software Effective? A Call for Empirical Research on Effectiveness. *Educational Technology*, December 1986, *26*(12), 7-11.

What Teachers Had to Say About Using Computers in Today's Schools. *Learning 86*, March 1986, 48-52.