Jl. of Computers in Mathematics and Science Teaching (1995) **14**(4), 559-571

Integrating Technological Assistance in the Standard Curriculum in Order to Improve the Success Rate on the Math Portion of the Pre-Professional Skills Test (PPST)

JOHN E. SASSER
Mathematics and Applied Sciences Department
University of Cincinnati
Cincinnati, OH 45221, USA

Computer programming exercises were used to enhance the curriculum of second semester mathematics classes for prospective elementary teachers. The exercises emphasized the math concepts and skills tested by the Pre-professional Skills Test (PPST), a standardized examination required for teacher certification in 12 states. The effects of the exercises were measured by testing PPST math score differences between the experimental group and a control group, which received only the standard curriculum. The pre-test consisted of a series of questions modeled closely on PPST math questions, while the post-test consisted of the PPST itself. The results, statistically evaluated by the McNemar Test, show that the experimental group demonstrated a significantly higher success rate for the mathematics portion of the PPST than the control group ($p<.01$). Consequently, computer programming exercises are shown to be an effective tool for improving math skills and performance on the math portion of the PPST.

Twelve states require that prospective elementary educators achieve passing scores on the Preprofessional Skills Test (PPST) before being certified to teach. In 1992/93, one-fifth of the approximately 50,000 examinees taking the PPST for the first time failed to earn an acceptable score on the mathematics portion of the test. This suggests the need for modifications to the standard curriculum which would improve the math skills and PPST math scores of the students.

The PPST consists of three parts: Reading, Writing, and Mathematics (PPST Bulletin, 1990). In the mathematics section of the exam, students are given 50 minutes to complete 40 items. They are not permitted the use of calculators or any other materials during the test. The mathematics test is scored on a scale ranging from 150 to 190, and scored solely on the number of items answered correctly. Each certifying institution or agency sets its own minimum for a passing score. Among the twelve states, these minimal scores range from 169-175, with most falling in the range of 170-172. Generally, a student needs to answer approximately 22 items correctly in order to receive a score of 172.

The questions on the math portion of the PPST are selected from different areas including: Arithmetic, elementary algebra, geometry, measurement, graph and chart reading. The math section test the student's ability to use his or her cumulative knowledge of mathematics and his or her reasoning ability. Computation is minimal; the student is not required to have memorized any specific formulas or equations.

Given the largely qualitative and conceptual nature of the mathematics portion of the PPST, any modifications to the standard curriculum should stress the mastery of concepts and the development of abstract reasoning skills. Writing computer programs to solve math problems can help refine student abilities in these areas—the act of breaking a problem down into a series of small, clear steps requires both mastery of the underlying concept and the application of logic to the problem solving process.

COMPUTER PROGRAMMING AND IMPROVING MATH SKILLS

A review of the literature reveals that studies relating the use of programming to improvement in mathematics skills goes back at least to the late 1960s. The first studies were concerned with the secondary level, generally grades 7-12. Wallace (1968) investigated the use of flow charts and computer algorithms in teaching high school trigonometry. He concluded that the use of these methods reinforced traditional teaching strategies and resulted in improved student learning. In a two year project involving five school systems, Haven (1970) investigated the use of programming and the computer as a problem-solving instrument. The results of this investigation showed that developing programming skills had a significant positive influence on mathematics achievement and abstract reasoning ability.

At the collegiate level, Fielder (1969) compared the achievement resulting from learning mathematical concepts by computer programming to the traditional class assignment approach. He found that computer programming increased student achievement and interest.

In general, research suggests that the same strategies and processes used in computer programming are those used by successful problem-solvers (Lewis, 1981; Milner, 1974). Finding and correcting errors in computer code—the process known as "debugging"—requires a careful rereading of the text of the program itself, and often a rethinking of the problem solving process. Mistakes can range from simple typographical errors to subtle, complex errors of logic.

Wells (1981) investigated this relationship between the processes involved in problem solving and the processes involved in computer programming. Using a clinical methodology, fifteen high school students attempted to solve 12 problems and to write 5 computer programs. The study provided evidence that the same processes used in computer programming are also those used in problem solving.

Sasser (1985) investigated the relationships between three methods of computer-assisted instruction in mathematics and the learning modality preferences of students. The three instructional methods—visual tutorial, visual-auditory tutorial, and programming problem-solving—were compared for effectiveness in improving algebra skills and attitudes towards mathematics. The programming method was found to be more effective over the entire sample than the tutorial methods in improving algebra scores. This finding suggests that the teaching of programming skills can be an effective way of helping students master basic mathematical skills and concepts.

In summary, the literature provides much support for the assertion that teaching students computer programming can also improve their cognitive skills and increase their achievement in mathematics.

In designing an educational strategy to take advantage of this correlation, it must be emphasized that writing and modifying computer programs provides larger increases in math skills when the problems and exercises to be programmed are directly related to the regular curriculum. Mastering basic mathematical concepts lies at the heart of most curricula, and the computer is ideally suited to provide practice and training in the techniques of learning these concepts.

The programming method of Computer Assisted Instruction (CAI) should be distinguished from CAI as it is usually implemented in the classroom—a graduated series of text and graphics used as tutorials and/or drill and practice.

The programming method of CAI requires students to write computer programs which solve mathematical problems. The computer functions as the “tutee.” The student, functioning as the “tutor,” then “teaches” the computer to solve problems and do mathematics. Billings (1983) suggested that when students write, test, and refine a computer program, they extend and apply concepts in new ways. The theory is that the writing of such programs teaches the learner a deeper understanding of the concepts involved. A variety of tools and techniques are learned in programming as students create series of instructions which express and implement these concepts.

It is this strategy which extends most the knowledge and mastery of mathematics by students. The efficacy of such a program is demonstrated in the experiment described below, in which the programming method of CAI was used to increase the math skills and PPST math scores of students seeking certification as elementary instructors.

EXPERIMENTAL

The Test Population

The population of students studied consisted of 102 volunteers: 96 females and 6 males in four sections of a mathematics course for elementary education majors. All 102 subjects were full-time students with an average age of 22 years (range=18-44). They had completed an average of 56.54 semester hours in college (range=13-141), with a mean grade point average of 2.75 (range=1.30-4.00). With the exception of one African-American male, they were all Caucasians. All graduated from high schools in West Virginia, Ohio, and Kentucky.

Definition of Control and Experimental Groups

Two of the four sections were selected at random to be the experimental classes, leaving the remaining sections to be the control classes. The two experimental classes consisted of a total of 52 students, and the two control classes had a combined enrollment of 50 students. The course lasted sixteen weeks. All four classes met an equal number of periods on the same days, and were taught by the same mathematics instructor.

Students received a syllabus indicating which chapters of the textbook would be studied during each class session. They were additionally informed which chapters would be covered by the mid-term test and which chapters would be covered by the final examination. Students were informed that all other tests would be announced a week ahead of time. The instructor prepared and graded all tests and homework assignments.

Pre-Test Procedure

Both the experimental classes and control classes were given a pre-test. The pre-test contained items which sought to test content knowledge in the same areas normally tested by the actual PPST, including graphs, fractions, percents, geometry, algebra, problem-solving, probability and number theory. (See Appendix A for sample pre-test questions.)

These baseline scores were used to verify homogeneity between the experimental and control classes at the beginning of the experiment, and also between the experimental and control groups in the statistical analysis of the experiment.

Treatment of the Experimental Group

During the first week of class, each student in the experimental group received a document explaining how to write simple programs in BASIC (Sasser, 1990). In addition, the use of the IBM-PC was demonstrated and the writing of simple programs was discussed.

Approximately 300 minutes of instructional time were built into 31 sessions distributed evenly throughout the course. These sessions were specifically devoted to writing programs that would solve problems similar to those on the pre-test.

The homework assignments, consisting of exercises from the text, had to be solved by writing, modifying or completing computer programs that would solve the exercises.

Treatment of the Control Group

Students in the control classes spent an equivalent amount of time (300 minutes) discussing and working problems similar to those on the pre-test. Regular homework assignments consisting of exercises from the text were assigned to be worked on paper in the traditional manner.

Regular homework assignments consisting of exercises from the text were given to both the experimental group and the control group. The only difference being that the students in the experimental group had to write, modify or complete computer programs that would solve the homework exercises. (See Appendix B for sample exercises.)

The Post-Test

Fifteen of the students in the control classes and twelve students in the experimental classes took the PPST following the semester during which the experiment was conducted. These students formed the final CONTROL GROUP and the EXPERIMENTAL GROUP, respectively, for the statistical analyses. At the particular institution where this experiment was conducted, students could choose when they took the PPST, as long as they passed it before graduation and certification to teach school. Generally, students would wait until their last year, or at least until after they had completed the required mathematics courses before taking the PPST. This is the primary reason why only 27 of the 102 students involved in the experiment registered to take the PPST on the scheduled testing date following the completion of the experiment.

RESULTS

The results of the experiment are summarized in Tables 1-3:

- Table 1 reports the pre-test and post-test scores for both the control and experimental groups and indicates (*) whether or not the score was passing.
- Table 2 summarizes the number and percentage of post-test scores that were passing and failing for both the control and experimental groups.
- Table 3 categorizes the experimental group scores on the basis of change. That is, instead of categorizing the same group of subjects, before and after, on whether or not they passed, the subjects are categorized on the basis of whether or not they changed from not having passed the pre-test to later having passed the post-test (the actual PPST). The change scores are independent of each other, and the sum of the values of the observed frequencies equals 12, the number in the experimental group.

Cell “a” contains only those subjects who changed from passing the pre-test to failing the post-test.There were none. Cell “b” contains only those subjects who did not change, passing both the pre-test and the post-test. Cell “c” contains only those subjects who did not change, from failing both the pre-test and the post-test, and cell “d” contains only those subjects who failed the pre-test, but passed the actual PPST.

From the data in Table 3 we use the McNemar equation: X^2cal = 9.00 > X^2.01(1)=6.64. Therefore, the success rate of students in the experimental group was significantly greater than the success rate of students in the control group (p>.01).

Table 1
Raw Test Score Data

Control Group		Experimental Group	
Pre-test	**Post-test**	**Pre-test**	**Post-test**
161	165	168	174*
166	167	171	180*
169	179*	166	172*
167	169	166	173*
180*	183*	168	178*
171	180*	167	178*
180*	182*	168	174*
178*	178*	170	175*
177*	177*	160	180*
168	173*	160	169
167	171	178*	180*
170	175*	173*	175*
180*	182*		
164	167		
163	167		
n = 15		n = 12	

* Passed (Score > 171)

Table 2
Pass Fail/Data

	Pass	Fail	Total
Control	9 (60%)	6 (40%)	15 (100%)
Experimental	11 (92%)	1 (8%)	12 (100%)
Total	20	7	27

Table 3
Changes in PPST Scores

		Post - test	
		fail	pass
Pre - test	pass	a o	b 2
	fail	c 1	d 9

DISCUSSION

The results of this study are in general agreement with research findings in the relevant literature—that the processes involved in computer programming are similar to the processes used in solving mathematical problems, and that computer programming exercises can be used as tools to enhance mathematical skills and performance on the standardized examinations used to measure them.

More specifically, computer programming appears to be an effective method for teaching the concepts and skills that regularly occur on the math portion of the PPST. This experiment suggest one way in which prospective educators in those states requiring minimum scores on the PPST might improve their success rate. Implementing programming exercises into the standard curriculum is no longer a daunting technical task. With the explosion of microcomputer technology in the 1980s, most schools now have adequate computational resources for this type of curriculum enhancement. Furthermore, older and more cryptic programming languages are slowly being replaced as teaching tools by prepackaged “scripting” languages, which have a closer resemblance to normal spoken and written language (Abelson & diSessa, 1981; Byte, 1982; Feurzeig, 1988). This type of software has become so adaptable that teachers and students are able to give simple commands to the computer for the manipulation of numbers or symbols, usually by “pointing and clicking” with a mouse, and the system automatically translates these actions into an easily readable and editable program. One example of this “programming by doing” is the scripting language HyperTalk (Informix, 1989). Other high-level language software packages designed to permit the user to easily construct programs are the Geometric Supposer (Sunburst, 1994), Boxer (diSessa & Abelson, 1986), and Function Machines (Feurzeig & Richards, 1991).

Finally, it seems likely that the programming method of CAI could be equally useful in teaching the general curriculum of mathematics taught in the elementary school, since the items normally found on the PPST reflect the same content. Given the current emphasis on increasing the computer literacy and technical skills of students at all levels, programming solutions to math problems could be introduced as an effective form of CAI in the mathematics classroom.

References

Abelson, H., & diSessa, A. (1981). *Turtle geometry: The computer as a medium for exploring mathematics*. Cambridge, MA: MIT Press.

Billings, K. (1983). Developing mathematical concepts with micro-computer activities. *Arithmetic Teacher, 30*(18-19), 57-58.

Byte. (1982). Logo and mathematics (Special Issue). *Byte 7*.

diSessa, A., & Abelson, H. (1986). Boxer: A reconstructable computer medium. *Communications of the ACM, 29*(9), 859-868.

Fielder, L.A. (1969). *A comparison of achievement resulting from learning mathematical concepts by computer programming versus class assignment approach*. Unpublished doctoral dissertation, The University of Oklahoma.

Feurzeig, W. (1988). *Apprentice tools: Students as practitioners*. In R. Nickerson & P. Zodhiates (Eds.), Technology in education: Looking toward 2020. Hillsdale, NJ: Lawrence Erlbaum.

Feurzeig, W., & Richards, J. (1991). *Function Machines* (Software). Cambridge, MA: BBN Labs.

Haven, R.N. (1970). *Project LOCAL-1970 evaluation*. (ERIC Document Reproduction Service No. ED 114 534).

Informix Software. (1989). *Wingz* (Software). Lawrenceville, KS.

Lewis, R. (1981). Education, computers and micro-electronics. *T.H.E. Journal, 3*(1).

Milner, S. (1974). *Learner-controlled computing: A description and rationale*. (ERIC Document Reproduction Service No. ED 092 157).

PPST Bulletin of Information. (1990). Oakland, CA: Educational Testing Service.

Sasser, J. (1985). *The use of computers as an integral part of instruction in mathematics*. Germany: Association For Supervision and Curriculum Development Yearbook.

Sasser, J. (1990). *Introducing LOGO and BASIC*. Marshall University Publications.
Sunburst Publications. (1994). *Geometric Supposer* (Software). Pleasantville, NY.
Wallace, D.C. (1968). *The impact of computer mathematics on the learning of high school trigonometry and physics*. Unpublished doctoral dissertation, University of Texas at Austin.
Wells, G.W. (1981). *The relationship between the processes involved in problem solving and the processes involved in computer programming*. Unpublished doctoral dissertation, University of Cincinnati.

Note

An early version of this paper was presented at the Joint Mathematics Meetings of The American Mathematics Society and The Mathematical Association of America (AMS/MAA), Cincinnati, Ohio, January, 1994.

APPENDIX A. PRE-TEST

SAMPLE PROBLEMS FROM THE PRE-TEST (AUTHOR CONSTRUCTED FACSIMILE OF THE MATH PORTION OF THE PPST)

Directions: MAKE NO MARKS ON THIS TEST! Each of the questions or incomplete statements below is followed by five suggested answers or completions. Select the best answer or completion of the five choices given and fill in the corresponding lettered space on the answer sheet.

.

.

8. Brian had a total of $34.18 when he went to The Koffie Bean. He purchased 6 pounds of coffee priced at $3.99 per pound. He gave the cashier $25.00 to pay for the coffee. How much money did Brian have after he purchased the coffee?

A.) $7.19
B.) $9.18
C.) $10.24
D.) $1.06
E.) $21.01

.
.
15. Given 2b=(10a + 2b)/2, what does “a” equal?

A.) 5/b
B.) 2/5
C.) 3b/5
D.) b/5
E.) 5b
.
.
40. The original price of a TV set was marked down 20% to the sales price of $440. What was the original price?

A.) $550
B.) $540
C.) $528
D.) $440
E.) $428

APPENDIX B

SAMPLES OF HOMEWORK ASSIGNMENT FOR EXPERIMENTAL GROUP

In this Appendix appears three programs written by students in the Experimental Group. These programs were selected because they correspond to the three sample questions in Appendix A:

TEST ITEM # 8

```
1 REM 04)T08.BAS
10 REM - TEST ITEM #8
20 REM - Store Purchases
30 KEY OFF:CLS
40 PRINT “What is the total amount of money you have”;:INPUT A
50 PRINT “DOES ANY ITEM SELL BY THE POUND (1=YESY;0=NO)”;:INPUT B
60 IF B=0 GOTO 110
```

```
70 IF B=1 GOTO 320
80 GOTO 50
90 PRINT
100 PRINT"We just figured the cost of all items that sell by the pound."
110 PRINT"We now wish to figure the cost of all other items you purchased."
120 PRINT
130 PRINT"How many items ";:INPUT N
140 FOR K=1 to N
150 PRINT"What is the cost of item ";K;:INPUT F
160 LET T=T+F
170 NEXT K
180 PRINT:PRINT "How much money was given the cashier ";:INPUT Q
190 REM - Change
200 LET R=Q-(T+T2)
210 PRINT:PRINT "Your change from the cashier should be";R; "dollars."
220 REM - Amount Left
230 LET S=A-(T+T2)
240 PRINT:PRINT "The amount of money you have left is ";A;" - ";T+T2;"=";S;"
        dollars":END
250 END
260 FOR K=1 TO N
270 PRINT"What is the cost of item";K;:INPUT F
280 LET T=T+F
290 PRINT
300 PRINT
310 PRINT"How much money was given the cashier ";:INPUT Q
320 PRINT"How many items sell by the pound ";:INPUT M
330 FOR L=1 TO M
340 PRINT"Cost per pound for item ";L;:INPUT C
350 PRINT"How many pounds did you purchase of item "L;:INPUT D
360 LET E=C*D
370 LET T2=T2+E
380 NEXT L
390 GOTO 90
```

TEST ITEM #15

```
1 REM 05)T15.BAS
10 REM - TEST ITEM #15
20 REM - Equation
30 KEY OFF:CLS
40 PRINT"This program solves equations of the following form:"
50 PRINT
60 PRINT"     By  +  Cx   "
```

```
70 PRINT"Ax = --------------"
80 PRINT"          D       "
90 PRINT
100 PRINT"What is the value of A ";:INPUT A
110 PRINT"What is the value of B ";:INPUT B
120 PRINT"What is the value of C ";:INPUT C
130 PRINT"What is the value of D ";:INPUT D
140 PRINT"Do you wish to solve for x or for y ";:INPUT A$
150 IF A$="x" GOTO 240
160 IF A$="X" GOTO 240
170 IF A$="y" GOTO 200
180 IF A$="Y" GOTO 200
190 GOTO 140
200 LOCATE 20,22:PRINT A;"X =";B/D;"Y +";C/D;"X"
210 LOCATE 21,22:PRINT B/D;"Y = ";A-(C/D);"X"
220 LOCATE 22,25:PRINT "Y = ";(A-(C/D))/(B/D);"X"
230 END
240 LOCATE 21,24:PRINT A;"X =";B/D;"Y +";C/D;"X"
250 LOCATE 22,26:PRINT"X =";(B/D)/(A-(C/D));"Y"
260 END
```

TEST ITEM #40

```
1 REM 06)T40.BAS
10 REM - TEST ITEM #40
20 KEY OFF:CLS
30 REM - Original Price (origprice)
40 PRINT"What is the percentage mark-down(express as a decimal)";:INPUT A
50 PRINT"What is the selling price";:INPUT B
60 LET D=1-A
70 LET C=B/D
80 LET E=1000*C
90 LET F=INT(E)/1000
100 LET F1=F/100
120 PRINT"The original price was";F1
```