DOCUMENT RESUME

ED 087 424 IR 000 168

AUTHOR Conord, A. E.
TITLE How The Computer Can Bring The Teacher and Student Closer Together.
PUB DATE Apr 73
NOTE 13p.; Paper presented at the Association for Educational Data Systems Annual Convention (New Orleans, Louisiana, April 16 through 19, 1973)

EDRS PRICE MF-$0.65 HC-$3.29
DESCRIPTORS *Computer Assisted Instruction; Computer Oriented Programs; *Computers; Computer Science; French; *Humanization; Speeches; *Student Teacher Relationship; Undergraduate Study
IDENTIFIERS AEDS; Association for Educational Data Systems; United States Naval Academy

ABSTRACT

Two computer-assisted courses taught at the United States Naval Academy, one in French and one in Computer Science, illustrate the ability of the computer to help teachers and students develop closer, more humanized relationships. In both courses the computer's basic role is to amplify the instructor's capabilities. Relying on its ability to process data at a high speed, the computer can be employed to store information, present assignments, conduct evaluations and provide immediate feedback. In addition to motivating students and helping them to diagnose their own needs, this last function also makes it possible for the instructor to modify his instructional plans in accordance with the progress to date of his students. Since the machine also relieves the teacher of much routine work it therefore provides him with the time as well as the information needed to design better instruction, thus promoting closer teacher-student ties. (PB)

HOW THE COMPUTER CAN BRING THE TEACHER AND STUDENT CLOSER TOGETHER

A. E. Conord

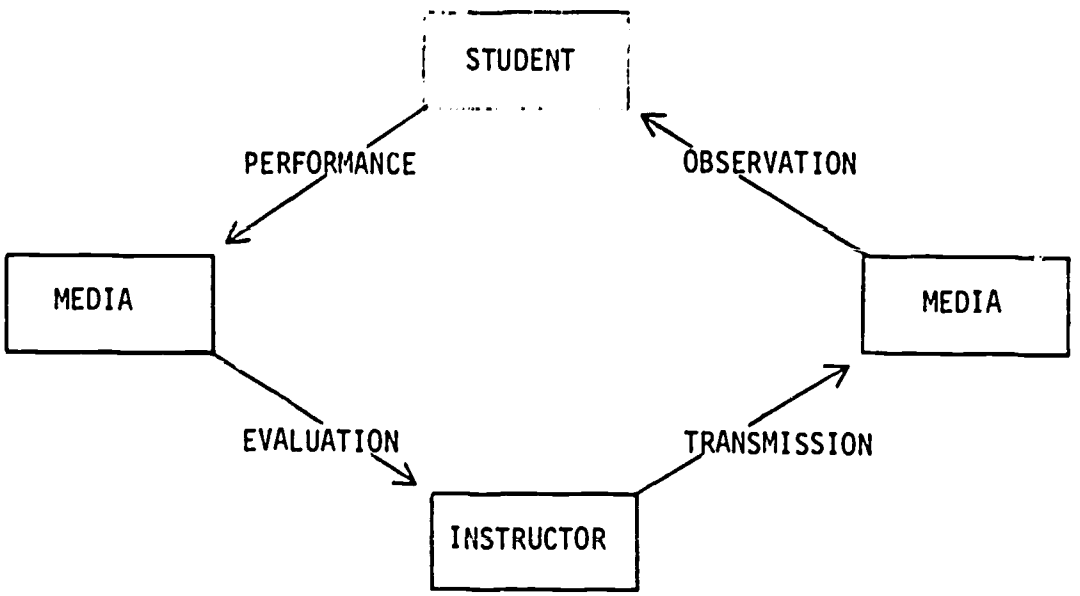
United States Naval Academy

INTRODUCTION

If at this point you are skeptical concerning the validity of the title, it is the author's hope that by the conclusion of this paper you will have become at least a qualified believer. At the outset let us bear in mind that the word "can" in the title can be replaced by "will" only if the teacher recognizes the significance of the dimension added by the computer and exploits it. Further, it should be mentioned that the two courses offered here as examples were selected because they represent courses actively employing two variations of the technique to be discussed -- not because they are to be considered the last word in this application. It is left to the fertile minds of the readers to explore new avenues and to innovate.

THE TEACHING/LEARNING CYCLE

The classical teaching/learning cycle finds the instructor transmitting information to the student via some medium. At a later time the student will be required to demonstrate what he has observed. Still later his performance will be evaluated, completing the cycle (Figure 1). The time lag between



CLASSICAL TEACH/LEARN MODEL

Figure 1

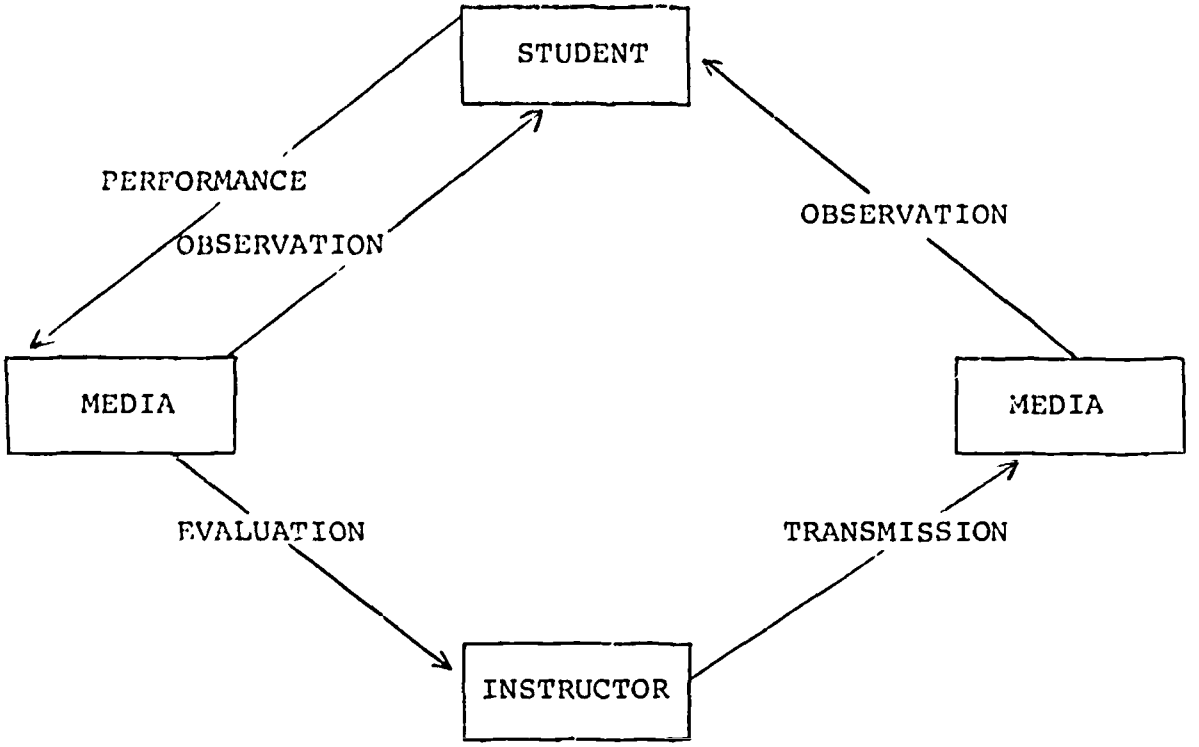
ALBERT E. CONORD is currently Director of the U.S. Naval Academy Academic Computing Center. He has a B.S. from the Naval Academy, an M.E.E. from Catholic University, and an M.S. (Computer Science) from the University of Maryland.

instruction and evaluation varies considerably, with the only consistent fact being that rarely does either the instructor or the student benefit from delayed feedback on the student's performance. The instructor might benefit if he teaches that course at a later date. But we are considering here what the value would be to teacher or student of having immediate student performance feedback. Given a model in which the student-to-teacher ratio is one, the manner in which immediate feedback for both student and teacher takes place becomes clearer and its value is apparent. What the instructor transmits to the student is influenced by the student's performance, which in turn resulted from his previous observation (Figure 1). The cycle is self-perpetuating, with negligible delay from cycle to cycle. However, when the student-teacher ratio is increased to two, a measurable delay is introduced since the instructor must now analyze the performance of both students before proceeding. Additionally, his reactive or reinforcing instruction may become too complex or cumbersome in an effort to satisfy both students.

As the number of students increases, the delays increase, and the ideal model begins to deteriorate to a point where there is no practical way for the model to function with timely feedback in either direction. An attempt to force this feedback might find the teacher spending a disproportionate part of the learning period conducting some form of testing, thus allowing less and less time for the transmission of new material. A mechanical analog of this model might produce some interesting responses as the parameter representing the number of students increases. Nevertheless, without belaboring this topic further, suffice it to say that instruction should be influenced by student performance which depends on timely performance feedback to both student and teacher. This is virtually impossible to achieve in the classical educational setting unless time can be conquered.

To uphold the thesis expressed in the title, it would seem imprudent to tamper with either the basic role of the student or that of the teacher. This leaves the media as the point of attack. Various forms of computer-assisted instruction tend to modify, even if indirectly, the instructor's function by way of the teaching media. Although he remains the creator, the teacher's role through CAI becomes transparent to the student. And even though it could be pointed out that CAI often incorporates the techniques to be set forth in this paper, it would hardly do to offer CAI as sole support of its thesis when many critics have branded CAI as dehumanizing. Instead, let us employ the computer in a way that enhances the human relationship between teacher and student. To do this, attention will be focused on the performance-evaluation branch of the model in Figure 1.

Figure 2 modifies the teach-learn model of Figure 1 to provide the student with the evaluation of his performance. Also the medium in this branch is acknowledged to be a computer. It is implied that the student-computer interface requires both input and output, with the teacher receiving only output. Although these modifications do not elevate the effectiveness of the model to that achieved in the earlier one-to-one model, it does offer significant improvement. First, each student enjoys immediate evaluation of his efforts, affording him the opportunity to correct any misconceptions he might have derived from previous observations. Second,



MODIFIED TEACH/LEARN MODEL

Figure 2

the instructor can now address his students at their next encounter forearmed with a precise analysis of both their individual and collective performances. To return to an early statement, his instruction is influenced by student performance. Recognizing that, conceptually, CAI provides these same advantages, CAI also employs the computer as the primary and often the sole instruction medium. The proposal being made here goes beyond intrinsic CAI by ignoring the transmission or teaching branch of the model. The instructor thus has the freedom of presenting his material as he chooses, using CAI or not.

THE ROLE OF THE COMPUTER

Educational psychologists and technologists have been wrestling with the problem of identifying optimal teaching methods. Regardless of how new information is presented to man, the depth of his learning often, if not always, is enhanced as he employs this information in some practical way. "Practice makes perfect" would support this philosophy. Homework has been a method long-used to instill confidence in the student regarding the utilization of the material to which he has been exposed. Assuming boldly that the student is motivated to make an honest effort to complete an assignment, one or more of several situations may result. To identify three should make the point: (1) the student may be confident of his

understanding of the assignment but in fact has some significant misconceptions, (2) he may have correctly completed the assignment but suffers a lack of confidence in his performance, or (3) a will-o-the-wisp instructor is unavailable when the problem is current and interest dies with the question unanswered. Whichever the case, the objectives of the assignment may, unbeknownst to both teacher and student, not be fulfilled. If the motivation assumption is renounced, the paradox of assigning homework is heightened. The computer can serve the instructor in solving this dilemma, and further it can enrich the technique of out-of-class work.

The computer's role in teacher-student relationship is manifold. First, it is the vehicle by which the assignment is presented. Moreover it can easily provide a different assignment to every participant who wishes additional exercise. Second, and more important, the computer provides immediate reinforcement to the student and a repository for any student response data desired by the instructor. Third, the instructor is provided student performance analysis in time to influence his next instructional meeting with the students. The computer's role is simply to amplify the teacher's capabilities. Its sole unique characteristic is speed. It does nothing that the instructor could not do if he had the time to do it. But, by extending the teacher's skills through the conquest of time, a new and potent dimension to teaching is created through the foreknowledge of student performance. Without deviating from his classic role of lecturer, if that be his wish, the instructor is armed with information heretofore unavailable.

Before looking at this technique in action, let's touch on a possible by-product of this approach -- student motivation. Does the computer motivate when thus employed? It is believed that most students would rather be gently reinforced when performing than not. To know that his deficiencies will be addressed and corrected at the next class session would also be an encouragement.

This has been an effort to present a concept that could improve teacher-student relations. It should be obvious that this concept is not rigid, that it imposes no restrictions on the teacher. To the contrary, when it comes to implementing the actual details, the computer should respond only to the wishes of the individual instructor.

THE COMPUTER IN THE HUMANITIES

Beyond the reason for supporting the theme of this paper, the course to be described offers an interesting challenge to the humanities instructor who has access to a time-shared computer. The professor who introduced the computer to his Introductory French course at the Naval Academy knew nothing of computers. He had seen a language grammar drill presented via remote computer terminal and set out to develop a complete set of drills directly linked to his text. His objective was to extend the grammar drills beyond classroom time by providing the student additional, yet supervised, exposure to each grammar lesson as it was presented. He developed and wrote the programs using a modular approach that minimized repetitive programming and one that would not be limited to the textbook he happened to then be using.[1] The student was provided reinforcement

[1] Further details on this course may be obtained through the author.

to his responses; but the instructor, in his inexperience, failed initially to realize that a potential base of important raw data that had always existed could now be captured. For instance, statistically, what are the most common grammar mistakes: Are they grammatical or spelling errors? Figure 3 is a computer output that represents a sample of an incorrect response analysis which answers these questions. From this information, two important actions emanate: (1) the computer programs can be updated to identify the difference between spelling and grammatical errors for the student, and (2) the instructor is forearmed with the grammatical weaknesses of the students.

This example is a straightforward application of the computer's role to extend the teacher's skill. The approach is readily adaptable to any course employing some uniformity in homework assignments where identifiable responses are expected.

ERR-ANAL 20 MAR 73 14:19

INPUT THE LESSON, AND EXERCISE? 30,IV

*****ERROR-RESPONSE ANALYSIS*****
OF
** LESSON #30 EXERCISE # IV **

GRAM-30
***INSTRUCTIONS:
REMPLACEZ "MEME SI" ET L'INDICATIF PAR "BIEN QUE" ET LE PASSE COMPOSE DU SUBJONCTIF:

QUEST.#	RESPONSE	FREQUENCY
1	AIT PLU	13
	BIEN QU'IL AIT PLUS	4
	BIEN QU'IL AITPLU	1
	BIEN QU'IL ATIT PLU	6
2	BEIN QUE JEAN AIT EU	9
	BIEN QUE AIT EU	1
	BIEN QUE AIT EUE	1
	BIEN QUE AIT UE	7
	BIEN QUE JEAN AIT EUE	3
3	HENRI SE SOIT TROMPE	1
	HENRY SE SOIT TROMPE	1
4	BIEN QU'IL JEAN AIT ACHETE	12
	BIEN QUE JEAN AIE ACHETE	3
	BIEN QUE JEAN AIT CHETE	1
5	BIEN QU'IL AIT DEJA VO@U	7
	BIEN QUE JEAN AIT DEJA VU	9
	BIEN QUIL AIT DEJA VU	17

INCORRECT RESPONSE ANALYSIS

Figure 3

It is unfortunate in one sense that the second example is a course in computer programming. Unfortunate because one is tempted to say, "What's so unique about using a computer in a computer programming course?" On the other hand it affords the opportunity not only to demonstrate the computer concepts described earlier but to introduce a means of monitoring student-written computer programs.

The Naval Academy requires that all first-year students take an introductory course in computer programming. The enrollment is therefore large (over 1000). As is generally the case in computer programming courses, the emphasis is on programming assignments rather than lengthy lectures. This affords the opportunity to increase the number of students per section but at the same time increases the administrative load on the instructors, such as having to correct 240 computer programs each week. Besides this problem, another one fundamental to beginning programmers existed. There are two basic types of errors a programmer can commit: (1) coding or syntactic errors, and (2) logical errors. The first type are easily handled since all modern compilers respond immediately to syntactic error with some form of diagnostic statement. The logical error is more insidious even to the experienced programmer. Once all syntactic errors are eliminated, how does the neophyte know if his program is delivering the correct answer? And if he is in error, how can he receive immediate help? For simple exercises he may have knowledge of the correct answer, but generally he does not. Further, with student-generated programs there is no unique program solution. This would appear then to eliminate the capability of reinforcing the student's efforts at solving the problem, and makes it difficult to automate the collection and analysis of performance data. If this be true, a large segment of the educational spectrum which employs the computer as a problem solving device is eliminated from the computer-modified teach/learn model of Figure 2.

Dr. John Kemeny, co-developer of the BASIC language, has augmented that language[2] with a software module called TEACH. TEACH provides the instructor with the means to analyze student-developed programs, provide reinforcement, and collect student response data. Using a very simple problem (Figure 4) as an example, let us follow the sequence of events that occur, describing the student, instructor, and computer responsibilities. The student is left free to attack the problem in his own way, subject only to some basic guidelines needed to provide communications with the auxiliary software, which is completely transparent to the student. Notice the restrictions indicated in Figure 4. For instance, specific variables are required to identify the results, in this case the list A(J) and the sum S. In addition, the student must name his program SIMPLST1, as indicated. There are no restrictions on the algorithmic approach taken by the student nor on any other variables he may desire to use. Having written and debugged his program, the student is ready to find out whether or not it delivers the correct answers. Upon typing the command TEST, the TEACH software transfers control of the student's program (Figure 5) to another program, which is generally written by the instructor. This

[2] Page 127, BASIC Sixth Edition, edited by Waite and Mather, University Press of New England, Hanover, New Hampshire, 1971.

NAME OF PROBLEM: SIMPLST1

PURPOSE: To teach calculations involving single subscripted variables.

MATHEMATICS INVOLVED: Write a program to read in a list A having 12 elements from a DATA statement. After you have read the numbers, find the sum of the terms S and then put the average in the 13th position. Then count how many numbers were below the average and put the result in the 14th position. Then print the entire new A with appropriate headings.

DATA: A(J) $1 \leq J \leq 12$

ANSWERS: A(J) $1 \leq J \leq 14$ where A(13) is the average and A(14) is the number below the average.
S = Sum of the elements of A.

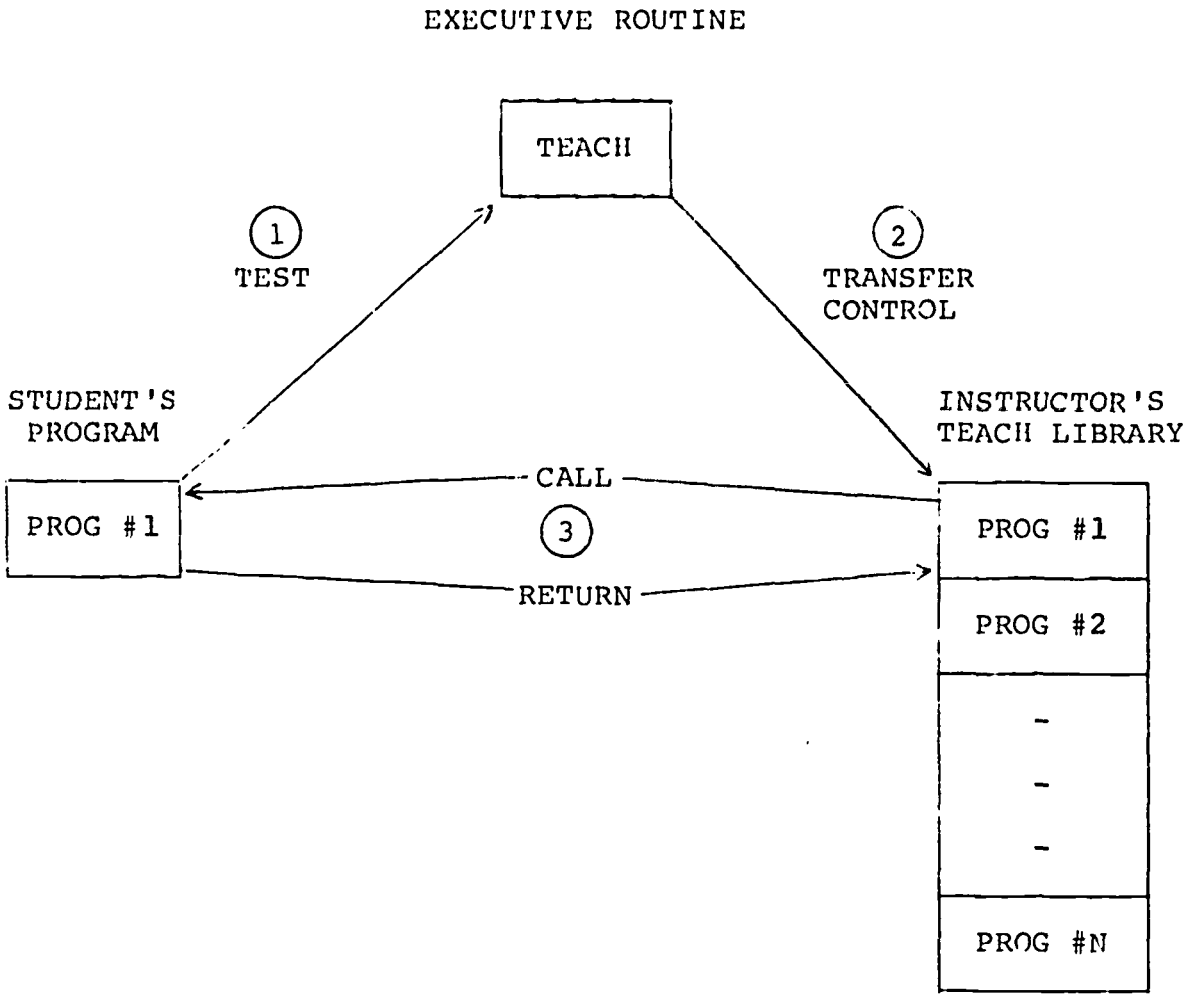
SAMPLE STUDENT ASSIGNMENT

Figure 4

instructor-designed control program calls the student's program as a subroutine and executes it, employing instructor-provided test data. Employing this data, the student's program will return anticipated results to the control program. Based on the results, reinforcement is fed back to the student. If the response was incorrect, the reinforcement might appear as in Figure 6. How much or how little help is, of course, the prerogative of the instructor, as it the option of assigning a grade. A correct solution is rewarded as described in Figure 7. The certificate of satisfactory completion shown here is provided the student in the event of a computer crash in which some data were lost. Complete documentation of the exercise consists of this certificate plus a "RUN" and "LIST" of the student's program.

Figure 8 is a skeleton TEACH control program, one of which is required for each specific assigned problem. The only required statements are underlined here. All other statements are remarks describing various options. V1, V2, V3, . . . are the mandatory variables prescribed in the problem instructions to the student. The complexity of the control program is dependent entirely on the number of anticipated incorrect results for which the instructor desires to test and on how elaborate is the reinforcement incorporated. Experience has proved that constructing control programs is not time-consuming, and only minor changes are required from one problem to the next.

What has been somewhat sketchily presented here is a means of providing immediate feedback to student-composed programs. Of course TEACH is not without limitations. There is a limit to the overall problem complexity. Further, the instructor must assign basically the same problem to each



STUDENT REINFORCEMENT OPERATING SEQUENCE

Figure 5

TEST

SIMPLST1 14 MAR 72 08:06

TEST IN PROGRESS

YOU ARE ASSIGNING THE NUMBER ABOVE THE AVERAGE TO A(14). YOU HAVE IT BACKWARD.

YOU HAVE RECEIVED PARTIAL CREDIT. TO IMPROVE YOUR GRADE, TRY AGAIN, PLEASE.

0.518 SEC. 44 I/O
READY

REINFORCEMENT TO INCORRECT RESPONSE

Figure 6

TEST

SIMPLST1 14 MAR 72 08:07

TEST IN PROGRESS

HURRAY. THIS PROGRAM WORKS. I THINK IT'S TIME TO 'RUN'
AND 'LIST' THIS SUBSCRIPT PROBLEM.

TEAR ON DASHED LINE

- -

YOUR GRADE HAS BEEN RECORDED. SIGN THIS AND SAVE IT AS A
BACKUP IN CASE OF A COMPUTER FAILURE. WHEN REQUESTED, TURN
IT IN TO YOUR INSTRUCTOR.

DATE: 03/14/72

TIME: 08:07:05

GRADE: 0200710199

- -

NAME

- -

TEAR ON DASHED LINE

DON'T FORGET TO 'LIST' AND 'RUN' YOUR PROGRAM.

0.550 SEC. 44 I/O
READY

REWARD FOR CORRECT RESPONSE

Figure 7

```
TEASKELT    08 MAR 72    11:54

100 ' TEACH SKELETON PROGRAM
110 ' AUTHOR'S NAME, DATE
120 ' NAME OF ASSIGNMENT
130 ' DESCRIPTION OF PROBLEM & SPECIAL FEATURES
140
200 ' LIST MANDATORY VARIABLES
210 TEACH V1, V2, V3....
220
300 ' SET TIME LIMIT (SECONDS) TO PREVENT RUNAWAY LOOPS
310 TIME 1
320
400 ' INITIALIZE MANDATORY VARIABLES TO RECOGNIZE IMPROPER USE
410
500 ' CALL STUDENT'S PROGRAM & SUPPLY A SET OF TEST DATA
510 CALL "PROG NAME":  V1,V2,V3....
520   DATA ...
540 ' TEST STUDENT'S PROGRAM FOR ANTICIPATED ERRORS
550
900 ' COLLECT STUDENT RUN DATA
910 '     (1) STUDENT I.D.
920 '     (2) EXERCISE I.E.
930 '     (3) ERROR CODE
940       (4) DATE/TIME
1000 END
READY
```

SKELETON TEACH CONTROL PROGRAM

Figure 8

```
/OLD AECSTUD***/LIST

AECSTUD          14 MAR 72       08:11

5708410196       02/04/72        09:15:18
5708410197       02/04/72        09:17:35
5708410199       02/04/72        09:19:42
5692310198       02/04/72        14:20:03
5346510196       02/04/72        15:09:49
5346510199       02/04/72        15:11:18
```

RAW STUDENT TEST DATA

Figure 9

RUN

STUDTAB2 14 MAR 72 08:53

EXERCISE #? 131

STUDENT ERRORS TABLE
*****FOR EXERCISE 131*****

08:53:14 03/14/72

NAME	68	88	96	97	98	99		
AVERYT D			*			*	03/08/72	17:14:02
BLASE R E						*	03/08/72	20:21:06
CASEY K D		*			*			
EICHELBERGER R		*				*	03/08/72	09:40:52
EVERET W M			*			*	03/08/72	16:30:34
FLIPPIN W N			*			*	03/02/72	13:40:11
GELHAUS F L			*			*	03/04/72	13:40:48
GRIBBLE R						*	02/27/72	17:00:47
GRIFFIN G B	*		*			*	02/22/72	14:35:23
HILL C A						*	03/09/72	18:00:10
TOTALS	1	2	5	0	1	9		

4.675 SEC. 58 I/O
READY

STUDENT PERFORMANCE ANALYSIS

Figure 10

RUN

STUDTAB1 14 MAR 72 08:50

STUDENT-EXERCISE TABLE

08:50:12 03/14/72

NAME	101	111	121	131	141	151	161	171
AVERYT D	*	*	*	*				
BAFUS J D								
BAILEY C E			*					
BANNON M C	*	*	*					
BLASE R E	*	*	*	*				
BREIDENTHAL J	*	*	*					
BUCKIEWCZ B	*	*	*					
BUTTERMOE J	*	*	*					
CASEY K D	*	*	*					
CONRAD L		*						
CORBETT F E		*	*					
DALEY W H	*	*	*					
EDGAR L I	*	*	*					
EICHELBERGER R			*	*				
EVERET W M	*	*	*	*				
TOTAL	10	12	13	4	0	0	0	0

6.359 SEC. 52 I/O
READY

STUDENT EXERCISE SUMMARY

Figure 11

student or be faced with the task of writing a separate control program for each student. However, this technique has application in any subject where large numbers of students employ the computer to solve the same basic problems.

Referring again to Figure 8, lines 900-940, it is shown that in addition to student feedback the control program can collect raw student RUN data such as his identification, date, time, and error codes representing specified incorrect responses. This raw information is collected in a file in a form shown in Figure 9. The ten-digit number contains the student identification, the exercise number, and an error response code. Additionally, the date and time the student completed the assignment is recorded. This raw data is reformatted and output to the instructor on demand, as shown in Figure 10. The two-digit numbers across the top are code numbers identifying specific errors anticipated from this exercise. Notice that both individual performance and error analysis are provided by this report. It should be evident that this technique eliminates the need for the instructor to collect and correct large numbers of programs, a tedious task at best. A by-product of the TEACH approach is shown in Figure 11. This table represents a running account of exercises completed. Not until an assignment is correct will a star appear in the appropriate exercise column for the prescribed student.

SUMMARY

The two examples presented both illustrate a contrast in the role of the computer from the student's point of view and in the possible disciplinary employment. It is important to realize that the computer can function as either a problem solving device or a drilling/testing medium while still serving to extend instructor capabilities by conquering time. Although the examples represent specific courses, one should not lose sight of the flexibility allotted the instructor. His imagination and creativity are necessary to extract from and embellish the concepts presented here. If the computer can give the student support when he most desires and needs it and, at the same time, arm the instructor with heretofore inaccessible performance information (which is extremely important) ... can it not, in the hands of imaginative and innovative teachers, serve to draw them closer to their students?