DOCUMENT RESUME

ED 098 957 IR 001 336

AUTHOR Boillot, Michel
TITLE Computer Communication Modes and Their Effect on Student Attitudes Towards Programming.
INSTITUTION Nova Univ., Fort Lauderdale, Fla.
PUB DATE Apr 74
NOTE 31p.; Practicum in partial fulfillment of requirement for Ed. D. degree

EDRS PRICE MF-$0.75 HC-$1.85 PLUS POSTAGE
DESCRIPTORS College Students; *Computer Oriented Programs; *Computers; Computer Science; *Computer Science Education; Input Output Devices; *Programing; *Student Attitudes
IDENTIFIERS Batch Process; Conversational Mode

ABSTRACT

The effect of different communicating modes to computers on students' attitudes toward programing was studied. In a computer-related course, 13 students used batch processing mode to solve problems on the computer, while 12 other students used conversational mode to solve the same problems. It was found that those students accessing the computer in conversational mode developed more positive attitudes toward programing (as measured by the author's attitude test) than the students using batch processing. Also the grades obtained in the time-sharing environment did not seem to affect the students' final attitudes towards programing. The majority of two-year college computer centers operate under batch processing mode. This could be a factor in the high student attrition rate in introductory programing classes. To lower this rate, and to make the computer more accessible to the campus classroom, it is suggested that minisystems with numerous learner consoles be used. (Author)

COMPUTER COMMUNICATION MODES AND THEIR EFFECT ON STUDENT ATTITUDES TOWARDS PROGRAMMING

APPLIED EDUCATIONAL RESEARCH
AND EVALUATION

by

Michel Boillot, M.S.
Pensacola Junior College

Cluster Coordinator: Herbert Vandort
Pensacola Cluster

A PRACTICUM PRESENTED TO NOVA UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF EDUCATION

NOVA UNIVERSITY

April 1974

TABLE OF CONTENTS

Abstract

The purpose of this study aimed at determining whether different communicating modes to computers could affect students' attitudes towards programming. In a computer-related course, thirteen students used batch processing mode to solve problems on the computer, while twelve other students used conversational mode to solve the same problems. It was found that those students accessing the computer in conversational mode developed more positive attitudes towards programming (as measured by the author's attitude test) than the students using batch processing. Also the grades obtained in the time-sharing environment did not seem to affect the students' final attitudes towards programming.

The majority of two-year college computer centers operate under batch processing mode. This could be a factor in the high student attrition rate in introductory programming classes. To lower this rate, and to make the computer more accessible to the campus classroom, the author suggests the use of mini systems with numerous learner consoles.

INTRODUCTION

Context of the problem

During the last few years the technology and the use of computers has undergone a continuous series of major changes. The degree of change and the speed with which it has occured has placed unprecedented pressures on Junior Colleges to keep pace. Recent developments of time-sharing in the computer industry have yet to find their way to the educational scene in most two-year institutions of higher learning (Journal of Data Education). The majority of Florida's 28 Junior College computer centers, for example, still rely heavily on batch processing as the only instructional avenue to the computer. Research and demonstrated projects related to computer support of instruction (project IMPACT) indicate time-sharing can become an extremely potent preceptive tool in an educational environment. Professor Kemeny of Dartmouth College suggests that cognitive and affective objectives may be attained more readily through the use of conversational communication with the computer than through batch processing mode.

Statement of the problem

The purpose of this study was to determine to what extent a student's attitude towards programming might be affected by the genre of communication with the computer. Would conversational use of a computer prove to be a better conduit to affective learning than batch processing and would grades obtained in programming courses affect in any way the learner's attitude towards programming?

Review of the literature

American education has failed to emphasize affective goals. B. O. Smith (1966) remarks that "...to teach any concept, principle or theory, is to teach not only for its comprehension but also for an attitude toward it." Many pedagogical

approaches to instruction are deemed successful when they produce acceptable cognitive outcomes; yet these same approaches often destroy any positive feelings a student might have towards a subject area. Mager (1968) points out that learning is for the future, i.e. that the object of instruction is to facilitate some form of behavior at a point after the instruction has been completed. "The likelihood of a student putting his knowledge to use is influenced by his attitude for or against the subject. Things disliked tend to be forgotten." Studies made by Myers (1950) have shown that positive attitude towards subject matter increases cognitive learning outcomes and results in higher student achievement. Hence the instructor, in addition to his role of dispenser of knowledge, should also be striving to discover new ways to promote affective learning. This implies some form of behavior change on the part of the student, teacher, or both. Oliver (1968) has shown that teachers will alter their behavior as the outcome of receiving feedback from their students. Changes in humanistic behavior of student teachers were also observed as a result of feedback (Tuckman, 1973). Hypothetically, then, one might infer that such results could also apply to student behaviors in a time-sharing environment. The very nature of a computer conversational operating system lends itself ideally to Thorndike's law of effect or Hull's principle of reinforcement (1943). The dynamic interchange between student and instructional system as a result of instantaneous feedback provides the learner with positive reinforcement. Studies on computer-assisted instruction in inquiry or game and simulation mode (Stolurow, 1968) also seem to indicate a gradual change of behavior on the part of the participant as he becomes more involved and interested in the process of interaction with the computer.

Hypothesis

The following hypotheses were formulated:

(a) Students in Introductory Programming courses will develop more positive attitudes towards computer programming when student communication with the computer is carried out in conversational mode as opposed to batch processing mode.

(b) Grades earned in such courses will not affect a student's attitude towards programming, that is a student can earn a grade of C in the course and still score as high on the attitu est towards programming (see Appendix A) as a student earning a grade of A.

Rationale

Thurstone (1946) defines an attitude as the degree of positive or negative affect associated with some psychological object. By a psychological object Thurstone means a person, institution, machine or idea with which people can differ with respect to positive or negative affect. Computers fit the role of psychological objects extremely well when used as an instructional tool by students for problem solving purposes.

Traditionally computers have always elicited from the student user either powerful approach responses (favorable attitudes) or avoidance responses (negative attitudes) depending on whether or not the student and the computer are provided with a suitable time period for mutual and gradual acquaintanceship. If this period of courtship is properly supervised by the instructor, students will generally develop strong approach responses to the computer. The reasons for this are manyfold. The computer, to the beginning learner, represents a novelty, an intriguing machine never heretofore encountered in his past experience, and therefore is likely to excite his curiosity. Secondly, the student can think of the computer as an ally or companion in a learning adventure; with the help of this ally the student can solve problems. Thirdly, and most importantly, the computer

provides the student with feedback in the form of listings of programs, diagnosis of errors if any, and results. In the event of program errors, beginning learners will typically display ambivalent sentiments towards the machine, first a sense of amazement that a machine might find something wrong with their product (creation), and secondly a sense of slight frustration at the machine's inability to read their mind. This in turn accentuates the desire of the student to demonstrate who really controls whom. In psychological parlance, the unconscious desire to triumph over or conquer the machine makes the learner even more tenacious or persistent in his desire for success. In the event of no program errors, the learner will tend to congratulate himself or his achievement, and a feeling of pride will overcome him, making his world a little brighter. This results in a temporary increase in self-esteem and confidence, enduring until his next confrontation with the machine. The same psychological loop of feelings is then experienced anew. Eventually these iterated experiences build upon one another to cement a solid positive relationship between the learner and the computer to the point where the student wishes more contact with the computer in order to experience this cycle of pleasant events. Affective learning has then truly taken place and the computer has endeared itself to man.

To a certain extent, the manifestation of positive feelings are induced by the computer irrespective of the communication mode used to interact with the system. Nevertheless, batch processing mode, by the very nature of its operational structure, may dampen student approach responses to the machine. Programming languages, such as FORTRAN, used in batch processing environment on small-sized computers may initially be confounding to the student because of the generality of the language and the numerous syntactical rules. Diagnostic error messages printed out by such languages generally prove to be quite difficult for the beginning learner to understand. Aversive conditions may also be

generated as a result of the delay experienced by the student waiting for his output. Students may wait hours for their programs only to find that their programs were cancelled due to one foolish keypunch error. Contact conditions with computer operators or lab assistants may also result in a display of animosity; programs can be misplaced, lost, shuffled, stolen, not processed, etc. All in all a host of negative factors over which the student has no control can occur which can be instrumental in decreasing his degree of approach responses to the computer. In a time-sharing setting the above mentioned problems are nonexistent. Programming languages such as BASIC were developed with the beginning student in mind; the language is designed for simplicity of comprehension, clarity of error diagnosis, and ease of operation. Also, in the time-sharing environment, the acquaintanceship period between student and console is minimal. Students immediately enter into dialogue with the computer and terminate when the problem is solved. Helpful error messages are printed, and corrections are made with no loss of time. When all syntactical errors have been corrected, results are then printed. In the event of logical errors in the program, the learner can ask the computer to print out values of variables and intermediary results. This ability to interact with the system, without any loss of time, is perhaps one of the most salient features of conversational mode. The computer seems to be continuously dedicated to the learner, and the learner in turn appreciates this mark of civility. Thus a rapport between man and machine is rapidly established.

As a result of the inherent structure of time-sharing, substantive changes in the potential of individual students can take place. Because the student can modify his program while at the console and change parameters, he can ask "What if" questions, and obtain instantaneous responses to a whole range of premeditated or spontaneous questions. He can mold his program as he progresses in order to

follow a changing train of inquiry. This fosters creativity and opens the way to independent study and research in a subject. The student may then have been sufficiently stimulated so that learning becomes an intrinsic self-generating drive. The scaffolding of the time-sharing strategy reveals the four levels of internalization in the affective learning process: receiving, responding, valuing, and characterizing.

Operational restatement of the hypothes

It was therefore hypothesized that conversational use of a computer would provide a more conducive climate to programming instruction than batch processing. The level of receptiveness and acceptability of the student to the computer would then be used to define operationally the student's attitude towards programming. Such an attitude could then be measured by an attitude scale test. In addition, in view of the anticipated impact of a time-sharing system over a learner's frame of mind, it was further surmised that grades would minimally, if at all, affect the learner's final attitude towards programming.

Significance of the study

Approximately 600 students at Pensacola Junior College annually enroll in the computer science course entitled "Introduction to Computers". The student attrition rate in such classes is close to 40%. The reason for this unusually high drop-out rate can be in part attributed to the use of batch processing by students. The relevance of this study to educational settings lies in the fact that time-sharing would seem to be more conducive to programming instruction than batch processing in terms of cognitive and affective outcomes. Time-sharing fosters learning by discovery which provides for greater transfer of concepts and principles

(Gagné, 1968). Transfer and transferability are clearly goals of the educational process (Tuckman, 1972).

METHOD

Subjects

The subjects were freshmen students at Pensacola Junior College who were enrolled in the computer science course entitled "Computers and Problem-Solving." All subjects in this intact group had previously taken "Introduction to Computers," and all were familiar with the batch processing mode, having written six or seven FORTRAN programs in the introductory course. All Ss were AA degree candidates and most expected to continue their studies at the local community university. Half of the intact group consisted of computer science majors. Fifteen students were randomly selected for assignment to the experimental group, and the remaining fifteen were assigned to the control group.

Independent variables

In the first hypothesis the independent variables were batch processing mode versus conversational mode. Both are Type A operational variables (Tuckman, 1972, pp. 58-59). These two variables describe two different modes of access to a computer. In a batch processing environment, various programs are collected into groups or batches, and then processed serially on the computer. In a time-sharing environment, the user has direct access to the computer through the use of a remote terminal.

In the second hypothesis, the independent variables consisted of students having earned a grade of B or above in "Computers and Problem-Solving" versus those having earned a grade of C or less.

Dependent variables

The dependent variable was student attitudes towards programming. This is a Type C operational definition (Tuckman, 1972, pp. 61-62). An attitude test was designed to measure Attitudes Towards Programming (ATP Test, see Appendix B). The

attitude rating scale consisted initially of 26 items each gauging the learner's attitude towards programming across the four levels of internalization described by Kratwohl and Bloom (1964).

1.	Receiving.	Sensitivity to the existence of stimuli or phenomena and the willingness to receive or attend to them. (Lewy, 1966, pp. 57-58)
2.	Responding.	Behavior which goes beyond the attention to the phenomena described in 1.
3.	Valuing.	Ability to attribute worth to the phenomenn and act in support of this value.
4.	Characterizing.	Ability to operate automatically, autonomously, and in a consistently integrated way.

For example item 5 asks whether the student would like to write some programs for some of his other courses. This establishes a value and a commitment to programming. Each scale has five points labeled: definitely not, probably not, undecided, probably, and definitely. As a result of testing for validity, a revised ATP test (20 items) was constructed in a shorter and more conceptually compact measure.

Reliability of the ATP Test

The internal consistency of the ATP was established by a test and retest procedure. Correlation of scores obtained by 20 students enrolled in Mathematics for Computers (a different course than Computers and Problem-Solving, used in this study) on the same ATP test given on two separate occasions (time interval 2 months) yielded a correlation coefficient $r = .65$, thereby making r significant at level $\alpha = 0.01$.

Validity of the ATP Test

Test validity was established as follows:

1. The ATP was submitted to a panel of experts (3 colleagues). Six items were eliminated as a result of the panel's collective judgment.

2. The ATP was then given to a group of graduating computer science majors, and to a group with a history of avoidance responses to computers. Variance analysis of the resulting ATP attitude scores revealed that the scores of the graduating computer science majors all reflected definite positive feelings towards programming while the other group's scores reflected a negative attitude towards programming.

Procedures

Students in Computers and Problem-Solving were randomly assigned to the experimental and control groups at the beginning of the semester. Ss were not told they were part of a research experiment. To minimize sensitization towards attitudinal testing, no pretests were given. In the course of the semester both control and experimental groups were given identical instructions and assignments, a total of 16 programs to solve various types of scientific and business problems. The experimental group solved their problems on a NOVA time-sharing computer using BASIC as a conversational language to the system, while the control group used IBM's FORTRAN language to access a System 360 IBM model 30 DOS operating system.

In the penultimate class meeting of the semester, students were given their final examination for the course. Then on the last day of class, both groups were asked to respond to the ATP questionnaire and both were told they would be advised of their final grades immediately after completing the ATP. In this way the students understood that their responses to the ATP could in no way affect their final grades.

Data Analysis

The hypotheses presented in an earlier section of this paper were tested statistically by use of t tests.

RESULTS

RESULTS

Hypothesis one: The results of t tests showed that conversational access to a computer did have a more significant effect on student attitudes toward programming than batch processing mode (df = 23; t = 2.46; $p < .05$). See Table 1.

Table 1. t test analysis of effects of computer communication modes (batch/conversational) on student attitudes toward programming.

Access modes	Mean	Variances	df	t	p
Batch (N = 13)*	3.31	0.634	23	2.46	0.05
Conversational (N = 12)*	3.95	0.198			

*Originally the entire group consisted of 30 students; by the end of the semester, however, 5 had withdrawn.

Hypothesis two: The results of t tests revealed that grades obtained by students using conversational access to a computer did not in any measurable way affect their attitude towards programming (i.e., students with low grades still had positive feelings towards programming). (df = 10; t = 1.258; the hypothesis that the means are equal cannot be rejected at the 0.05 level.) See Table 2.

Table 2. t test analysis of effects of grades on attitude towards programming.

Grade distribution	Mean	Variance	df	t
B or above (N = 7)	4.121	0.306	10	1.258
Below B (N = 5)	3.78	0.077		

DISCUSSION

Conclusions

It should be noted that the conclusions of this study are based on an experiment involving twenty-five participants. In this context, then, and in terms of affective learning, conversational mode proves to be a more preferable instructional media for programming instruction than batch processing. Its catalytic effect on student learning coupled with its heuristic potential makes conversational mode a powerful pedagogical tool, with such clout that negative psychological reactions toward the computer do not seem to take place even when low grades are obtained. This in itself is of paramount significance, for in the course of their continuing education students who have scored low in freshman programming courses will not automatically shy away from subsequent computer-related courses, nor will they have to fight mental blocks when confronted with research work requiring computer use.

Discussion

Introduction to Computers is a 3-credit course offered at the Pensacola Junior College, designed to introduce students to the world of computers and their applications. Of the 600 students annually enrolled in this course, approximately 240 students obtain a D, F, or receive no credit. These figures, indicating a sizable failure, can be attributed to a certain extent to the computer system configuration of the college. Similar problems also exist in other colleges, for in fact batch processing mode still remains the prevalent mode of communications with a computer in most two year educational institutions. Ironically enough, this may be the least desirable configuration to have for instructional purposes in introductory courses to computers. It is somewhat inefficient, time-consuming, and initially confounding to the students (if not properly guided) when compared to

a time-sharing system. The primary educational responsibility of a computer science department in a two year college is to provide a computer system whereby any beginning student can interact easily with the system at no loss of time. Access to a computer should be an exercise in serendipity, and therefore efforts should be made to select a computer system, or to include additional systems (mini-systems perhaps) which would encourage approach responses on the part of students. Failure to the student is not so much the fact that he receives little or no credit in return for his payment of fees, or that it delays his graduation, but the impact of his confrontation with his failure, in other words how he internalizes the consequences of his failure. Failure may generate a host of psychological problems which can severely affect his attitude toward education, self, and life goal: feelings of incompetence or inadequacy, lowering of self esteem, anxiety, etc. It leads to negative approach responses to the subject matter and/or formation of mental blocks. To the college, and indirectly to society, failure represents resources misspent in that no learning took place.

Recommendations

High failure rates in introductory programming courses at junior colleges could be minimized by the introduction of time-sharing systems. There are now literally dozens of computer manufacturers marketing small-size time-sharing computer systems (mini computer systems) allowing up to 16 or 32 learner communication terminals. These can be purchased at the cost of five or six month's rental price of regular medium-size computers. Of course, medium-size computers can also operate in time-sharing mode, but at prohibitive costs to junior colleges when numerous inquiry terminals are included in the system configuration. Mini computer systems on the campus scene would not sound the death knoll of large or medium-size computers. The latter remain indispensable for processing the large

volume of business and administrative paperwork of the college and for instructional purposes to train students majoring in data processing or computer science.

For introductory programming courses, however, or for other courses such as mathematics, physics, engineering, business, etc., the medium-size computer operating in batch processing mode cannot be used successfully as an adjunct educational tool to supplement or complement instruction in the classroom. The time required for mastery of the language, the accessibility, and the affective learning process can best be provided by time-sharing or conversational systems. Until the medium or large-size computer can dissipate the shroud of secrecy which surrounds it (operations, communications, and complexity of use) the computing center will never fulfill its role in servicing and meeting the educational needs of students. The computer center falls short of its goals when only professional programmers and computer science majors can readily access the computer. The computer should be brought into the classroom, and not vice-versa. This can only be achieved by means of conversational terminals. The computer should be all-encompassing in its preceptive reach on the campus; sadly enough, this is rarely the case. This points to a serious defect in the policy, philosophy, or statement of purpose of the two year college computer center.

Epilogue

Concurrently with this study, the author of this report conducted a semester-long in-service project aimed at familiarizing faculty members with the computer center, the computer and its use. Faculty members came from various departments such as mathematics, social science, astronomy, adult high education, and business administration. Numerous programs were written both on the medium-size computer (batch processing mode) and on the mini computer (conversational mode). At the conclusion of the in-service program, it was unanimously agreed that conversational

mode was by far the superior pedagogical tool.

APPENDICES

APPENDIX A

Table A-1. Effect of computer communication mode on student attitude towards programming.
The entries in the table reflect the mean obtained by each student across all 20 items of the ATTP test. Intact group: 25; Batch group 13; Conversational group 12.

Mode of Communication

Batch	Conversational
2.75	4.15
2.95	3.70
3.00	3.55
3.10	4.45
4.50	4.20
4.75	3.95
2.50	3.80
3.35	4.00
4.00	3.00
3.05	3.65
2.00	4.00
3.35	4.35
4.10	

Table A-2. Effect of grades on student attitude towards programming in a conversational environment.
The entries in the table reflect the mean obtained by each student across all 20 items of the ATTP test. Intact group 12; above B 7; below B 5.

Grade distribution

Below B	B or above
3.70	4.50
3.55	4.45
4.20	3.95
3.80	4.00
3.65	3.00
	4.60
	4.35

02b

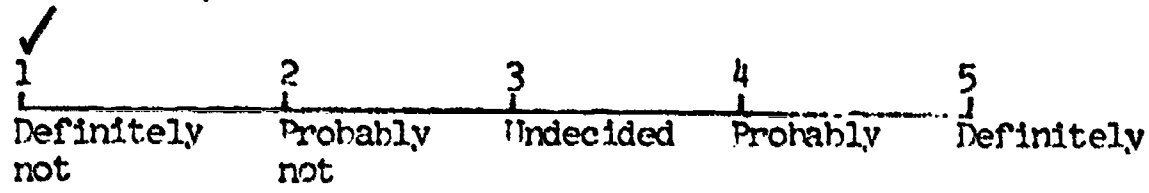
APPENDIX B

Attitude Test Toward Programming. (ATTP)

Test instructions: Please indicate your opinion by making a mark on the scale above the word you choose.

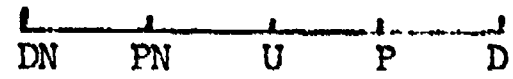
Example:

Do you like fishing?

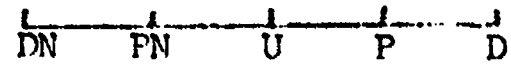


In the following scales DN, PN, U, P and D respectively mean Definitely not, Probably not, Undecided, Probably, and Definitely.

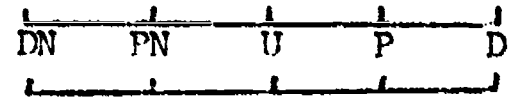
1. Do you find programming enjoyable?



2. Would you look forward to a job which would entail some programming responsibilities on your part?

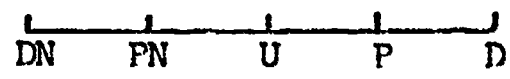


3. Will this course affect your decision in the selection of your major?

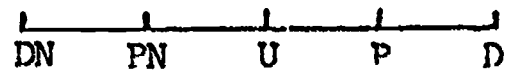


4. Would you recommend this course to someone?

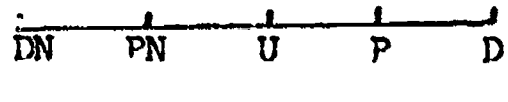
5. Would you like to write some programs for some of your other courses?



6. Given the time, would you presently be interested in joining a club which meets to discuss programming?

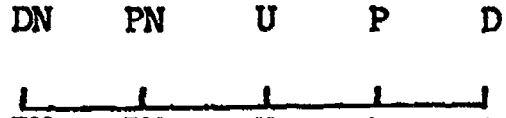


7. Do you wish you had more time to devote to programming?



8. Can you work on a computer program as long as an hour without being bored?

9. Do you try sometimes to convince some of your friends to become interested in data processing?

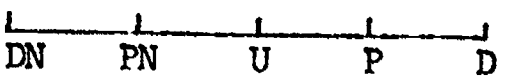


10. Do you often make it a point to read data processing articles?

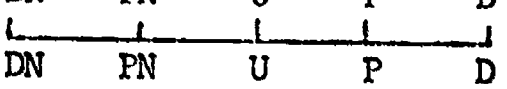
11. Do you agree that the word "elegant " is suitable for describing a specific computer program?

12. No music nor poem, no painting, no other work of art gives me a greater sense of the beauty of harmony than a well developed computer program.

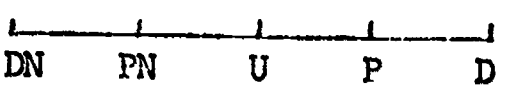
Have you ever experienced a sense of beauty in computer programming as described in this statement?



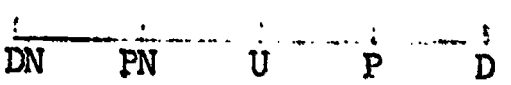
13. Would you take this course again?



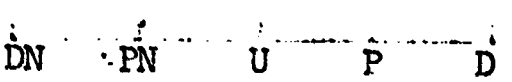
14. Do you feel you have been sufficiently motivated to consider learning an intrinsic self generating drive?



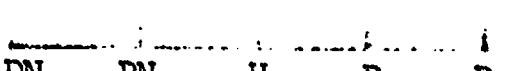
15. Has this course left you with a positive attitude towards programming?



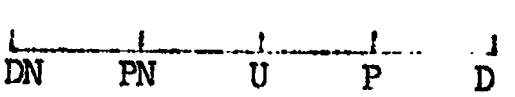
16. Do you feel your curiosity has been excited?



17. Are you now more predisposed to programming than you were before taking this course?

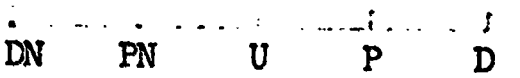


18. Do you feel programming is more of a game than it is hard work?



19. Do you enjoy programming more than you enjoy other courses?

20. After you have started on a program are you more interested in finishing it than doing almost anything else?



REFERENCES

REFERENCES

Bloom, B.S. and others. Handbook on formative and summative evaluation of student learning. New York: McGraw-Hill, 1971.

Edwards, Alan. Techniques of attitude scale construction. New York: Appleton-Century-Crofts, 1957.

Gagné, R.M. Learning hierarchies. Educational Psychologist, 1968, 6(1), 1-9.

Hull, C.L. Principles of behavior. New York: Appleton-Century-Crofts, 1943.

Kratwohl, D.R., Bloom, B.S., and Masia, B.B. Taxonomy of educational objectives: The classroom of educational goals. Handbook 2. Affective domain. New York: McKay, 1964.

Lewy, A. The empirical validity of major properties of a taxonomy of affective educational objectives. Unpublished doctoral dissertation, University of Chicago, 1966.

Mager, R.F. Developing attitude toward learning. Palo Alto: Fearon, 1968.

Myers, R.C. and Schultz, D.G. Predicting academic achievement with a new attitude interest questionnaire. Journal of Educational and Psychological Measurement. X(1950), 654-663.

Smith, B.O. Teaching and testing values. In Proceedings of the 1965 invitational conference on testing problems. Princeton, N.J.: Educational Testing Service, 1966. Pp. 33-49.

Society of Data Education. The misdirection of computer education efforts. Journal of Data Education. Reprint. Vermont: Northfield, 1970.

Stolurow, Lawrence M. Computer assisted instruction. Detroit: American Data Processing, Inc., 1968.

Thorndike, E.L. The psychology of learning. (Educational psychology, Vol. 2). New York: Teachers College, 1913.

Thurstone, L.L. Comment. American Journal of Sociology, 1946, 52, 39-50.

Tuckman, B.W. Conducting educational research. New York: Harcourt-Brace Jovanovich, 1972.

Tuckman, B.W. Effectiveness of feedback to teachers as a function of source. Journal of Educational Psychology, 1968, 59, 297-301.

Tuckman, B.W. and Walencik, V.J. The effectiveness of feedback for changing student teachers' humanistic behavior. Unpublished paper, Montclair State College.

U. S. Continental Army Command. Letter of instruction: Initiation of requirements for computerized educational system applications. August, 1969.

Seidel, Robert J. et al. Project IMPACT: Computer-administered instruction concepts and initial development. Tech Report 69-3. Alexandria, Virginia: Division No. 1 (Systems Operations), Human Resources Research Office, March, 1969.

Writer's Name [illegible]

Cluster PENSACOLA JUNIOR COLLEGE

I certify that I have read this practicum report and have discussed its contents with the writer.

[illegible]
(date)

[illegible]
(signature of peer reader)

I certify that I have read this practicum report and have discussed its contents with the writer.

Nov 1st 74
(date)

G. Mills
(signature of peer reader)

I certify that I have read this practicum report and have discussed its contents with the writer.

Nov 14 1974
(date)

P. Einbecker
(signature of peer reader)

.....

I certify that I have read this practicum report and that in my opinion it conforms to acceptable standards for practicums in the Doctor of Education Program.

(date)

(signature of Practicum Director)