INTERACTIVE TESTING PROGRAM

by

Scott Howard Stuart

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University in

partial fulfillment of the requirements for the degree of

MASTERS OF SCIENCE

in

Computer Science

APPROVED:

G. W. Gorsline, Chairman

R. B. Frary

B. L. Kurshan

July, 1982

Blacksburg, Virginia

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the members of the Department of Computer Science of Virginia Tech for their general support over the past two years, while I have been working towards this degree.

I would also like to thank Professor Robert B. Frary, for technical assistance on computer-assisted testing. His help on the statistical portions of the program was especially beneficial.

I would also like to thank Professor Barbara L. Kurshan of Hollins College. She provided me with unlimited use of the computing facilities at Hollins College. I benefited from frequent discussions with her, on many aspects of the program and paper.

Finally , a special thanks to Professor George W. Gorsline, my advisor, for his guidance throughout my graduate studies and the development of this paper. He has been very patient and understanding during this time. He has always been available to me, and has been especially flexible with his time.

CHAPTER 1 INTRODUCTION

CHAPTER 2 REVIEW OF LITERATURE

CHAPTER 3 QUESTION TYPES

CHAPTER 4 TESTING AND MEASUREMENT

CHAPTER 5 TECHNICAL IMPLEMENTATION

CHAPTER 1

INTRODUCTION

This paper will discuss the research, design, implementation and testing of an interactive testing program that allows: easy test entry and test generation, randomized selection of questions, on-line testing and grading, and statistical analysis of results. The program is written in VAX-BASIC, and is implemented on the VAX-11/780, running under the VMS operating system, at Hollins College, Hollins, Virginia.

Specific areas included in this paper include a review of the literature in the field of computer-assisted testing; a discussion of question types, and of testing and measurement; a summary of the technical implementation and operation of the program; a complete user's manual; and a discussion and conclusion.

In the review of literature, the area of computer-assisted test construction will be covered, with some detail discussion given to item banking, item generation, item selection, interactive testing, and adaptive testing.

Question types discussed include free response questions and response selection questions. In the chapter on testing and measurement, test and student statistics, as well as item analysis theory and methods are covered.

A test is a "device for obtaining a sample of an individual's behavior" [ALLEM79,p1]. In order that a test be useful is must accurately and consistently measure some property of that behavior. The testing program designed for this paper is designed to help a professor develop useful tests. The program is user-friendly for both professor and student. The design and implementation is well structured and completely modularized. The program is adequately documented both internally and externally.

In the technical discussion of the program, areas covered include naming conventions and file structures used; a list of the programs and subroutines along with their use; and the method of installation and execution of the program. A complete user's manual is included in this presentation.

The appendix has a brief summary of the user's manual.

For information on obtaining a copy of the programs and subroutines, both in printed and executable form, contact Barbara L. Kurshan, Director of Academic Computing, Hollins College, Hollins, Virginia.

CHAPTER 2

REVIEW OF LITERATURE

The field of computer-assisted test construction has been growing steadily in the past eight years. Areas discussed include item banking, item generation, item selection, interactive testing, adaptive testing, question types, and an overall view of computer-supported testing.

2.1 ITEM BANKING

Item banking is the assembling of a set of test items on a computer. These items are also selected by the computer. The method of selection can be random or it can be by a number of categorical methods.

While the idea is not a new one, there are many problems in the area. There are many diverse often overlapping efforts at gathering item banks and this has "led to few easily disseminated, generalizable products" [HISCM80, p3].

There are good reasons to use item banks. Test items are "even better after use" [HISCM80, p6]. Multiple use of the same item allows the gathering of item statistics from which conclusions about the validity of the particular item can be drawn. Nevertheless difficult distribution and lack of standardization have so far left most item banks at the local or regional level. "What prevents the effective use of items written by others is the lack of a method to consistently provide the test constructor with the right item--an item that measures the right skill at the right difficulty level with an appropriate stimulus, response format, and scoring technique." [HISCM80, p57]

2.1.1 BYU

In 1975 the second conference on CATC (Computer-Assisted Test Construction) was held in Atlanta. The largest system discussed was one at BYU in which over 135,000 tests were taken and scored during 1974. This system uses a large item bank from which the computer generates and scores the tests. For each question there are a number of correct as well as incorrect answers. One correct and four incorrect answers are chosen for each item. The tests are taken by hand, off-line on an opscan sheet. The student goes to a test center, identifies himself and is handed the appropriate test. He takes the test and then turns it in to be scored by a computer. The student receives his score soon after completing the test [WALLS75].

2.1.2 CIBEDS

The Minnesota State Department of Personnel uses a CATC system to administer over 300 different tests to more than 30,000 applicants yearly. They began to implement this system in 1975. All of the existing test items were classified according to the Dewey Decimal System by knowledge areas. This was initially done manually onto index cards. [STOCM80]

The system was computerized on a Control Data Cyber 74 at the University of Minnesota. The system was called Computerized Item-Banking and Exam Development System (CIBEDS) and was written in FORTRAN. This was done so that the system would be portable to newer computers in the future. While the system has several flaws it is much better than the manual system that had been used. It allows storing, formatting, and modifying items. Item statistics are generated and used to help upgrade the test. Items can be selected by content area and on specifications of their statistics. After the items are chosen by one of the above methods, relevant insertions such as reading passages are automatically added to the appropriate questions. A letter-quality copy of the test is produced. [STOCM80]

Most of the shortcomings of CIBEDS were caused by a lack of understanding of the system in the first stages of its design. These areas are being or have been improved. "At, present CIBEDS is operational and is used to create and print three to five new tests per week." [STOCM80,p14]

2.1.3 CAP

At the University of Alberta a computer-based test production model is in use in the medical and allied health fields. The system uses a Competency Analysis Profile (CAP), test item banking and retreival, and optical scanning capabilities. [SKAKE79]

CAP is used to define and describe the competencies that are to be tested via a two-dimensional matrix with rows designating competency categories and columns designating specific competencies within that category. Items in the test bank are designed to test specific competencies from the CAP. Test items must be entered by a prescribed format. One file contains the questions while a corresponding file contains the item statistics and other relevant information, including what competencies it tests. [SKAKE79]

Test production is done by a committee. Each member selects those competency categories on which he wants to test, and how many questions will come from each category. After each committee member receives a printout of the appropriate items for their selections, they jointly select the particular questions they want to include on the test. The program permits one to choose particular items by item number, and in that way the test is produced. [SKAKE79]

Once the test has been administered, it is scored by an optical scanner and the item statistics are automatically updated. Reports to the examinee include a diagnosis of his strengths and weaknesses according to the competency categories. Reports to the administrators of the program include an item anlysis and specific as well as summary student progress reports. [SKAKE79]

The developers of the system conclude that "the system provides error-free examinations at a favorable cost, supplies information about the test and the examinees, and above all improves the quality of the item bank by storing each item's performance for every time that the item is used." [SKAKE79, p710]

2.1.4 Discussion

The main benefits of item banking are a categorization of questions, and an analysis of how valid each item is. The item analysis is easily implemented in any system. Question categorization is a time-consuming and possibly subjective process, and often is not done well. Nevertheless, having a large easily accessible bank of questions should lead to better prepared, more accurate and more reliable tests.

2.2 ITEM GENERATION

One problem with item banking is that it requires a large amount of storage. There is another method of computer-assisted test construction that requires much less storage. This is item generation, where individual, unique test items are generated each time they are needed. Preset variables are given random values within certain predefined ranges. These values are substituted into the problem. The computer generates the solution to the problem. after it has been formed. One program or subroutine is written for each different type of problem that is included [COLLR79]. This requires more computation time and less storage. Given that computation costs are declining relative to storage costs, computer item generation should play a major role in the future [SUBRR75]. However, this type of problem also requires a large investment in development time.

2.2.1 PROBGEN

One such item generation program called PROBGEN (PROBlem GENerator) has been implemented at Eastern Michigan University. The designers, Collins and Duff, set out to design a system that would automatically generate problems, and one that required no programming skill to use. "PROBGEN prompts the user to provide the generalized solution to each problem, any major restrictions which might apply, the limits on all variables within each problem, and the exact wording for the text of the problem" [COLLR79,p115]. While the user need not be a programmer, he does need to thoroughly conceptualize the problem, and organize it such that it will fit into PROBGEN.

Producing each problem is a seven step process. These steps include : defining the problem and variables; arranging the necessary formulas; specifiying limits for subscripted and non-subscripted variables; selecting a FORTRAN compatible name for each variable (i.e. IGRADE for an integer variable for a grade); specifying invalid values or relationships among the variables; and composing the exact wording of the text of the question. After composing the questions, the instructor composes a heading for the test and selects how many problems of each type will be included. After approving a test run with solutions, he can get a hard copy of the test. [COLLR79]

As with most item generation systems, PROBGEN is only applicable to quantitative problems. Item generation can produce only a limited range of test problems and has genrally been used for scientific and mathematical test. Nevertheless, PROBGEN is accessible to the non-programmer, and it does generate individual, unique problems.

2.2.2 IPSIM

IPSIM is a system devleoped at the University of Iowa. The original system was able to "store, select, and print banked questions, and to administer tests at the terminal" [LIPPG75, p13]. Sophisticated item generation capabilities were added in 1975. This was done with the help of an authoring language delveloped for IPSIM. This system combines capabilities from many different areas of computer-assisted test construction.

2.2.3 Washinton Technical Institute

It has already been mentioned that most item generation capabilities involve quantitative questions. One system developed at the Washington Technical Institute by Olympia uses a new approach. Different item components are stored in several item banks. One component from each of several banks is chosen and then assembled into a complete question. This method of assembling text components is a novel approach that holds much promise. The above mentioned system also has the traditional item generation capabilities of generating random numeric data for predefined variables. [LIPPG75]

2.2.4 Discussion

Item generation is an interesting area of computer-assisted test construction. There is a need and room for further work, especially as computing costs come down. It provides for a virtually unlimited supply of unique questions. While storage capabilities are small, question develpment may require more time than it would using a banked system. However, the greatest restraint on item generating capabilities today is their predisposition to quantitative questions. There has been some work on other types of questions, however there is still much work to be done. It is easier to conceptualize the generating of numeric questions than that of non-numeric questions. As more work is done in artificial intelligence and syntactic text manipulation, more text generating systems should be developed.

2.3 ITEM SELECTION

Item selection must be done whether test banking or item generation is used. This task can vary from being very simple to being quite complex. The simplest system for item selection will involve using all questions that are available.

More complex themes involve chosing items by assigned and/or measured attributes. These attributes include key words, key topics, difficulty, popularity, and discrimination index [SUBRR75]. Most systems will only use a few attributes, however more complex systems can use a large number of attributes in order to choose the questions. Attributes can either be stored with the items or in seperate files. One system used by the Educational Testing Service, Princeton, New Jersey, stores only the attributes on-line and assembles the test manually after the items have been selected by the computer [SUBRR75].

Item selection also will involve the number of questions included on a particular test. This can be broken down to the number of questions from a particualar area or with certain attributes.

For multiple choice type questions, after the items have been chosen, the answers provided must also be chosen. This once again can range from a simple to complex operation. In the simplest case, all answers provided will be chosen. A more common and interesting method provides multiple right and wrong answers from which one (or more) right and one (or more) wrong answers are selected. For several reasons, one correct answer per item is the norm. Multiple correct answers lead to added overhead and a reduction in the effectiveness of optical scanner devices. Additionally, according to [FRARR79D], it has not been shown that including more than one correct answer is beneficial. In fact a second correct answer is often included by mistake.

Item selection can be quite simple or rather complex. It must however always be done whether one is using a computer-assisted or manual test construction method. Later, what types of questions should be used with computer-assisted test construction will be discussed.

2.4 INTERACTIVE TESTING

Computer-assisted test construction may or may not also include on-line or interactive testing. Bork defines interactive testing as a test, quiz, or exercise "taken by a student sitting directly at either a computer terminal or an individual (personal) computer system" [BORKA80, p2]. Areas discussed under interactive testing include overall design guidelines and the use of feedback. Reviews of several interactive testing systems are also presented.

2.4.1 Design Considerations

Designing an interactive testing system should be similar to designing any computer dialog system. "The student is holding a 'converstaion' with the writers of the program through the computer." [BORKA80, p2] Therefore all methods of designing user-friendly programs should be employed. Perhaps, in a testing environment, it is even more important to allay the fears a student might have of the computer. He will already be somewhat apprehensive about taking the test.

The program must also be designed so that the instructors using the system need not have extensive computer experience. The instructors should have no concern with the internal format of the question, only with how they need to interact with the computer in order to enter their questions. [BORK80]

2.4.2 Feedback

Feedback can come in a variety of forms. The spectrum of uses starts with no feedback. While it is true that providing feedback is not a difficult thing to implement on a computer, many professors may not want to provide the student with any information as to the correctness of their choice. However, if a professor wants to use the test as a learning tool, feedback is a great aid. Bork states that "the learning aspects of on-line quizzes are very important" [BORK80, p10].

The simplest type of feedback will be to simply tell the student whether the answer was right or wrong. A futher modification is to provide the correct answer if the student chooses the incorrect one. This is a rather simple matter, and provides a good amount of learning. Further refinements will require more sophisticated systems. These would call for returning the item to the question pool, and providing additional problems on the same topic for questions missed. [BORKA80]

2.4.3 LEVEL3

An interactive system designed for use in the medical field is in use at the School of Basic Medical Sciences of the University of Illinois, Urbana-Champaign (SBMS-UC). This system has been in use since 1971, and is used to teach and test topics from thirteen basic medical sciences that are studied in the preclinical years of medical education. The program has over 400 self-instructional units which are based on over 4000 learning objectives. [SORLW79]

Both the self-instructional packages and the comprehensive exams based on one of nine major areas are interactive. The examination system is called LEVEL3. The system "is written in TUTOR for PLATO IV as implemented on a CDC 7600" [SORLW79, p654]. The PLATO IV system is one of the most heavily funded CAI packages in the world. It has been in existence since 1959 and contains most aspects of CAI capabilities including a complete authoring language, TUTOR, as well as computer-assisted test construction and administration. [DENES78]

Each LEVEL3 exams contains about 180 questions and requires four to five hours to take. The primary objective of the exam is to improve learning, not to assign grades. This objective is accomplished by providing short-term goals, providing detailed diagnostic information to the student, and providing immediate feedback. [SORLW79]

In order to take an exam, the student interactively reserves a terminal and time period. He receives an immediate confirmation or denial. The student then goes to that terminal at the designated time. It is already logged-on and ready to begin the appropriate test. Each test covers several subject areas. The examinee may skip a question and either return to it later or receive a zero for it. Feedback is provided only after all questions in a discipline have been answered or skipped. At that point, "students may opt to review the questions answered incorrectly (recycle) or proceed to the next discipline" [SORLW79, p655]. During review, the student may change incorrect answers. After completion of the entire test, the student receives an on-line, detailed summary of his performance. [SORLW79]

Sorlie concludes that the advantages of the system "were not due solely to the computer-assisted medium, but to the process made possible by the medium: a 'teaching exam' with interactive review and immediate feedback" [SORLW79, p660].

2.4.4 MASTS

A Microcomputer-Assisted Study and Testing System (MASTS) is in use at the University of Texas. Objectives for MASTS include using disk-based microcomputers; making the system interactive and as far as possible transportable; including an authoring system and a statistical analysis. The system is implemented on a TRS-80 48K MODEL III computer. [GARRH80]

MASTS is contained on two disks, one for the authoring system, and one for each student containing data and index files. Each student gets a seperate student disk. The system is based on an instructional algorithm. The instructor develops the algorithm by chosing from among a menu-displayed list of options. Options allow the instructor to choose the type of question, the type of acceptable answers (either exact or by key word), and the type of information that will be gathered about the student. [GARRH80]

The student disk contains detailed information about the student as well as about his performance on each item. When a student finishes a test, he is given instructions about what he is to do next. At some point he is instructed to turn the diskette in to the instructor. When all disks have been turned in, the instructor can run the statistical program. This will produce a statistical summary about each test and about each student. [GARRH80]

In addition to the benefits for managing instruction, the MASTS project reduces the computer-related anxieties of the students using the system. Many of them are first-time computer users. [GARRH80]

2.4.5 Discussion

Interactive testing provides a logical extension of computer-assisted test construction. One of the main problems is the expense involved in establishing and maintaining the system. Microcomputers may provide an answer to this problem although obtaining a large number of microcomputers still can present quite an expense. Feedback is an easily implemented area of on-line testing and it provides a good opportunity for a 'learning test'.

There are a number of interactive testing sytems in use, some using microcomputers and others using larger interactive systems.

2.5 ADAPTIVE TESTING

Adaptive tests are also interactive. These tests are adaptive in that the next question asked is based on the answer or answers to previous question(s).

Adaptive testing algorithms are iterative and often consist of four steps : obtain an initial estimate of the student's ability level; use this to select the first question; administer and score that item and use that information to revise the estimate of the ability level; and determine when the ability level has been throughly refined [STOCM80].

2.5.1 CAT

Based on the work of Lord, an adaptive testing system called Computerized Adaptive Testing (CAT) was developed at the Educational Testing Service in Princeton, New Jersey. CAT is used experimentally in schools, with classroom terminals communicating with a DEC PDP-11 computer in Princeton. [SWANL80]

Lord demonstrated that "CAT, with only twenty-five items, is as effective in measuring verbal aptitude as a comparable 50-item conventional test" [SWANL80,p15]. Each test is tailored to the individual student, with subsequent items based on previous responses. After each response the computer evaluates the student's ability level based on all previous responses, and asks the next question from that level. Twenty-five items are asked with each item coming closer to the student's ability level. The scoring is based on the ability level reached, not on the number correct or incorrect. [SWANL80]

An advantage over traditional tests is that CAT can measure a wider range of performance. CAT measures verbal ability from the fourth-grade to the graduate level, using two parallel test forms of the Broad Range Tailored Test of Verbal Ability. [STOCM80]

Using the system is simple. All a student has to do is to press the correct letter in response to a question and a list of choices. All record keeping is done by the computer, thus freeing the teacher from that chore. Feedback as to the correctness of a response is provided and a student can retry a question that was answered incorrectly. [SWANL80]

2.5.2 Discussion

Computerized adaptive testing is a fairly new field but it hold much promise. It allows a broader range of measurement with a fewer number of questions. Developing and implementing such a system requires a lot of time and expense. An adaptive testing system is usually more difficult to develop than a less adaptive one. As more work is done in this area, more systems should be developed.

2.6 COMPUTER-SUPPORTED INDIVIDUALIZED TESTING

There are several problems in typical school settings which can be remedied by computer-assisted testing. These problems include the inability of providing for retesting; the inflexibility of the testing schedule; problems with cheating; use of class time for testing; large amounts of time spent in the preparation of a test; no item statistics; and large administrative expenses [CALHW78]. These problems lead to poorly constructed and administered tests. A well constructed computer-supported system might overcome these problems.

Calhoun suggests four features of such a system. The first is an item file that allows a number of correct and incorrect responses for each question stem. Outline material and item statistics would also be included in this file. Extra correct and incorrect choices reduce the need for redundant test items. The second feature is the ability to generate unique tests which measure the same abilities. This is accomplished by large item banks maintained by subject areas, using random selection. Calhoun advises that the item bank should be published. This requires a large enough bank so that memorization is not possible. This will reduce problems related to test security. The fourth point mentioned is that testing should be done at independent times outside of regular class time. [CALHW78]

Many of these features were written into the testing program produced for this paper. Some of these points are dependent on the method of implementation.

CHAPTER 3

QUESTION TYPES

3.1 FREE RESPONSE ITEMS

Free response items refer to "item types in which the student is required to generate a response rather than select from two or more provided responses" [THOMS81, p1]. Many people have a bias toward this type of question, beleiving that it better measures what a student knows. A recent study [THOMS81] indicates that this may not be so. Thompson states that ". . . there may be no basis in measurement reliability or validity for the prevailing bias towards FR (free response) items and against RS (response selection) items in quantitative disciplines" [THOMS81, p12].

The testing program designed for this paper includes a free response format for questions. Instructors may want to include this type of question especially for simple definitions or identifications. The program does no partial recognition so that the answer must be exactly correct. Therefore, one or two words answers would be best used under this program.

3.2 RESPONSE SELECTION ITEMS

Response selection items come in a variety of forms. All of these can be implemented on an interactive test. The forms that will be discussed are multiple choice, answer until correct (AUC), and concealed multiple choice.

3.2.1 Multiple Choice

One of the most widely used form of test questions is multiple choice. It also is one of the more controversial types. Concerns with this type of question are that it encourages güessing and rewards the wrong types of skills.

Many teachers feel that a student can get points for sheer guessing. This led to the introduction of scoring devices to negate the effect of guessing. However, there are some problems with such a system. It is rarely the case that someone is making a completely wild guess, but rather has eliminated one or more choices and is making a somewhat educated guess. Telling student 'not to guess' may be interpreted differently by different students. This will lead to more variability than guessing would. In fact, most scoring schemes do not penalize completely random guessing, but rather negate its effect. This means that on the average, an examinee has nothing to lose by guessing. [FRARR82]

While guessing may lead to some added variability in scores on multiple choice tests, the effect is often not as severe as many educators perceive it to be. It should not in itself cause an instructor to move away from multiple choice questions entirely.

Bork contends that multiple choice items on a test do not really measure what a student knows, but rather how well that student is at taking a multiple choice test; that is the ability to look at a set of answer, eliminate some, and guess among the remaining selections. He concludes that multiple choice "is almost never necessary in good on-line computer quizzes, except possibly for occasional variety of question form" [BORKA80,p8].

Garson makes a stronger case against all response selection questions. He says that 'drill and study' questions place the student in a passive role and tends to teach the student skills that are better done by the computer. He states that "the goal of computer design, then is not to figure out how to get the computer to play the role of the teacher, but to have it serve as much as possible as a resource for the student" [GARSJ80,p30]. He admits that it will be difficult to implement some of the changes he wants. How it is to be done, or even exactly what is to be done, is never made clear by the author.

Garson does admit that a computerized test does have the advantage of being easily modified based upon past performance [GARSJ80]. Multiple choice questions can measure a student's performance. This is especially true when the items have been repeatedly used and invalid items removed or modified.

Multiple choice items have been included in the testing program designed for this paper. This has been done for the above reason, and also because it is a familiar type of question and should be readily accepted by professors using the package.

3.2.2 AUC

Answer until correct is another type of respone selection question. Under AUC, the examinee is presented with all possible choices, just as he is under multiple choice. He is then instructed to select one choice at a time until the correct response is chosen. Scoring is based on the number of choices made with the best score being one choice for each question. This method provides immediate feedback in the form of the correct answer and for that reason does present a good learning opportunity. [FRARR80]

The simplest method of scoring AUC questions (and the one used in this program) is to award a point for each answer not chosen. For a discussion of a number of other scoring schemes see [FRARR81].

3.2.3 Concealed Multiple Choice

Bork recommends a question structure that was developed at the Irvine Educational Technology Center of the University of California, Irvine, by Franklin. It is concealed multiple choice. Under this method, the student is presented with the question stem, and one by one, a series of possible answers. For each answer, the student is to say whether or not it is a good answer. This process proceeds until the examinee makes a mistake (accepting a bad answer or rejecting a good answer). The advantage of this method is that the student does not see all answers at once and must make a decision about each choice. Different students will see different sets of responses. [BORKA80]

This does seem to be an interesting approach to interactive testing but is not included in this program. The aspect of random selection of answers, so that different students will see a different set of responses has been included.

3.2.4 Discussion

There is no single best type of question. There are advantages and problems with each. Each professor may have a preferred type. Both free response and a variety of response selection items have been discussed. In the program designed for this paper, free response items and two forms of response selection items have been included.

CHAPTER 4

TESTING AND MEASUREMENT

A test is a "device for obtaining a sample of an individual's behavior" [ALLEM79,p1]. In order that a test be useful is must accurately measure some property of that behavior. Measurement is the "assigning of number to individuals in a systematic way as a means of representing properties of the individuals" [ALLEM79,p2]. These numbers must be assigned in a consistent manner so that the same performance will receive the same number, regardless of other circumstances. [ALLEM79]

The measuremnt tools used in the testing program are designed to produce consistent test scores. How these scores are used in assigning grades is up to the professor. A discussion of the measurement tools included in the program follows.

4.1 TEST AND STUDENT STATISTICS

The most commonly used descriptive statistics for a set of scores are the central tendency and the variability of the set.

4.1.1 Mean

The central tendency of a set of scores "indicates where the center of the score distribution lies" [ALLEM79, p16]. In other words, what is a common grade. The three most common measures of central tendency are the mode (most freqeuntly occurring score), the median (the middle score), and mean (the arithemtic average of the scores) [ALLEM79]. The indicator included in this program is the mean. It has been given as a raw score as well as in a percentage format. The mean is a fairly stable indicator of central tendency. It is however more sensitive than the mode or median to a few very large or very small scores. [ALLEM79]

4.1.2 Standard Deviation

Varaibility of a set of scores "indicates the degree to which the scores vary" [ALLEM79, p16]. The three most common measures of variability are the range (difference between the largest and smallest score), the variance (the average squared deviation from the mean), and the standard deviation (the positive square root of the variance) [ALLEM79]. In the current program, the standard deviation is used to measure variability. The measure used is for a population, not a sample.

The standard deviation is a useful measure of how 'spread out' a particular set of scores are [FRARR79C]. The program accurately calculates this measure of variability.

4.1.3 T-score

In order to reduce the effects of great variability and to standardize scores from one test to another, T-scores have been included in the program. T-scores are standardized scores with a mean of 50 and a standard deviation of 10. They standardize for differences in difficulty level and variability across test. "T-scores report only a student's relative position in the class and are not an absolute measure of achievement" [FRARR79C, p1].

4.1.4 Assigning Letter Grades

How one defines letter grades is a professional decision. There are no rules that will answer all situations. Some observations and suggestions from the Virginia Tech Learning Resources Testing Advisory Committee include [FRARR79C] [FRARR79A] :

i) Determine a typical grade distribution for the course.

ii) Determine circumstances which might cause you to deviate from this norm.

iii) Identify the grades of students whose work you know. See if they come out with appropriate grades.

iv) Determine the naturally occurring gaps in the distribution.

v) Do not announce the correlation between score and letter grades before giving the test.

vi) The T-score spread need not be the same for each letter grade.

vii) There does not have to be an equal number of each letter grade assigned.

viii) The scores may not fit a 'normal' distribution.

4.2 ITEM ANALYSIS

Item analysis procedures are used to identify poor or invalid items. It may also offer information about how a poor item can be improved. Those items which have been determined to be poor should be discarded. [ALLEM79] Different statistics have been used depending on whether the test has been taken in 'regular response mode' or under 'answer until correct'.

4.2.1 Regular Response Mode

Regular respone mode applies to multiple choice type questions and is the standard method of administering and scoring the test. One answer per question is permitted. One point is given for each correct question. Some questions are 'keyed response'; that is, there is a question with no answers provided. The student is to enter the correct answer. The test permits up to 49 possible correct responses but does no partial word recognition. The answer must be exactly the one in the answer bank. These questions are scored like true/false-multiple choice items. All correct reponses are analyzed together and a dummy incorrect answer is put in to be analyzed for all incorrect responses. For regular reponse mode questions, four measures of the validity of each item have been included. All four are provided for correct as well as incorrect responses.

The first measure is the number of times the item was used on any test. The next measure is the percentage of the time that it was chosen. If the choice is correct, this measures the difficulty of the question. One would probably want most correct answers for a particular question to be of equal difficulty. If the choice is incorrect, this percent measures the attractiveness of this item as a distractor. Extreme values are undesireable. [FRARRR82]

The third measure provided is the average T-score of those selecting the choice. Since the mean T-score is 50, if the choice is correct one would want this measure to be above 50. If the choice is incorrect one would want this measure to be below 50. In other words the better students should be choosing the correct answers consistently. [FRARR82]

The fourth and final measure for regular response mode questions is the standard deviation of the T-scores of those selecting the choice. The standard deviation of all T-scores is 10. One wants this measure to be meaningfully below 10 since a smaller ability distribution of students should choose any one answer. [FRARR82]

4.2.2 AUC

For the testing program, the statistics kept for AUC questions are different for correct and incorrect choices. When the test is administered under AUC, all questions used will have four choices, and keyed response questions will not be chosen. For both correct and incorrect answers, the number of times the choice was used is provided.

For correct choices two more statistics are provided. The first is the average 'soonness' of selection. Right choices will always be chosen if they are used. Therefore what one wants to measure is how soon it is chosen. The lower this measure, the easier the item.

The last measure kept for correct answers is the correlation between 'soonness' and T-scores. This will be in the interval [1,-1]. A positive correlataion would indicate that higher scores (better students) tended to get the item right sooner than students with lower T-scores. A postive correlation is desirable.

For incorrect responses three further statistics are kept. They are the percent selecting the choice, the average T-score of those selecting the choice, and the standard deviation of the T-scores of those selecting the choice. These can be interpreted in a similar manner to the same statistics under regular response mode (see above). [FRARR82]

CHAPTER 5

TECHNICAL IMPLEMENTATION

The program is written in three main programs and six additonal files containing about 50 subroutines.

The three main programs that operate the package are TESTENTRY.BAS, TAKETEST.BAS, and REPORTER.BAS. TESTENTRY is used to enter a new test, revise an existing test, and to work on testing options. TAKETEST administers the test. REPORTER is used to compile the statistical summaries and item analysis. TESTENTRY and REPORTER are designed to be run by the professor and are password protected. TAKETEST is intended to be run by the student. It is only necessary that he know the professor's name and the test name in order to operate this program.

5.1 TESTNAME

The testname is a name of up to eight characters that begins with a letter. All of the subsequent information is stored in a file that is keyed to the test name, professor's name, and a special tag that indicates the purpose of that particular file. Each file name is made up of the test name, a one character code, and then a three character file type that consist of the first three letters of the professor's name. Codes '1','2', and '3' indicate that the file contains the questions and answers for tesbanks 1, 2, and 3. Corresponding to these are codes '4', '5', and '6' which indicate that the files are pointer files for testbanks 1, 2, and 3 respectively. An '0' in the code position indicates the file that contains the options for the test.

For example, for a test named 'RELATIVE' (RELATIVITY would be too long) for a professor EINSTEIN, testbank 1 would be called 'RELATIVE1.EIN'.

5.2 PASSWORD

The professor enters a password. This can be any set of up to twenty letters and numbers that begins with a letter. All passwords are converted to uppercase. There is also the option of using no password.

The password provides a level of security for entry into the test file, as well as the program that generates the student's statistics. Once a password is chosen there is no way to change it nor to review it by using the program.

The password is encoded and it is stored in the first record of the first test bank in the encoded form. When the professor is asked for his password when using an existing test, it is taken in without being printed on the screen, encoded by the same routine, and checked against the one in the file. The encoding subroutine is called ENCODE_PWD and is contained in TESTSUBS1. The scheme is based on a key which comes from the first letter of the professor's name.

5.3 QUESTION TYPES

There are two general types of question. These are multiple choice and keyed response.

Multiple choice question have a one or more line question stem, from one to 49 correct answer and one to 49 incorrect answers. Each answer can be one or more lines long. For any test, if the item is chosen, one correct and three or less incorrect answers are chosen. Less than three are used if there are less than three incorrect answers in the bank.

Keyed response questions have a one or more line question stem and from one to 49 single-line correct answers. A dummy incorrect answer is placed in the bank for item analysis. When the student enters an answer, it is checked against all entries in the bank and if it matches any one of the correct answer it is marked correct. It must match a correct answer exactly.

5.4 TEST BANKS

The package allows one, two, or three test banks with each bank containing from one to 100 questions. Typically the different banks will contain questions on material from different subject areas. A given number of questions on the test will be selected from each bank. Questions from the same bank will be grouped together on the test.

5.5 ORGANIZATION OF TEST BANK FILES

The file name for the test bank is the first eight characters of the test name plus the character '1', '2', or '3', depending on which test bank it is. The file type will be the first three letters of the professor's name.

Each record of a test bank consists of three informational fields, and six statistical counters. The first field is a key field, the second is a tag field, and the third is a buffer. The map statement of these relative files is :

```
MAP (TSTENT_MAP) KEY$=4%, BUFFER$=60%, TAG%,
                 N_XPOSED%, N_SELECTORS%, T_SELECTORS%,
                 T2_SELECTORS, T_XPOSED%, T2_XPOSED%
```

5.5.1 Key Field

The key field indicate what the buffer field contains. The key is a character string that contains up to four characters. The keys and their meaning are given below.

KEY	MEANING
QMC	first record of a multiple choice question
QKR	first record of a keyed response question
ANC	first record of a correct answer
ANI	first record of an incorrect answer
CQM	continuation of a multiple choice question
CQK	continuation of a keyed response question
CANC	continuation of a correct answer
CANI	continuation of an incorrect answer

5.5.2 Tag Field

The tag field is a unique six digit number that identifies the question as well as each answer. The leftmost digit will be a 1, 2 or 3 , corresponding to the number of the test bank. The next three digits from the left will indicate the number of the question within the test bank. The two rightmost digits will indicate the number of the answer for that question. Numbers less than 50 are reserved for correct answers and those from 51 to 99 are used for incorrect answers. Bank and question headers are given a tag of zero. The question stem has the last two digits as zeros, while the other information pertains to the bank and question number. (i.e. for test bank 2, question 45, correct answer 3, TAG = 204503, for incorrect answer 2, TAG = 204552)

5.5.3 Buffer Field

The first record of the test bank and the first record of each question will be a header record. In these two cases the buffer field contains the header information. Each entry in the header record is seperated by a colon. The last entry is indicated by two colons. Individual entries are extracted from the header by a subroutine called FINDNXT. If the buffer is not used for a header then it contains the text of the question or answer.

The first record of each test bank contains information about the test bank and has the following format ('+' indicates concatenation):

```
PASSWRD$ + PROFNAME$:FLNAME1$:Q.NUM$:B.RCDS$::
```

VARIABLE	USE
PASSWRD$	password
PROFNAME$	professor's name
FLNAME1$	name of file used for the test bank
Q.NUM$	total number of questions in bank
B.RCDS$	total number of records in the bank

The first record of each question contains information about that question. The buffer field of that record has the following format :

Q.MARK$:Q.NUM$:Q.RCDS$:AC.NUM$:AC.RCDS$:AI.NUM$:AI.RCDS$::

VARIABLE	USE
Q.MARK$	type of question
Q.NUM$	question number
Q.RCDS$	number of records for the question
AC.NUM%	number of correct answers
AC.RCDS$	number of records for correct answers
AI.NUM$	number of incorrect answers
AI.RCDS$	number of records for incorrect answers

5.5.4 Counters

The six statistical counters correspond to the answers and only those counters in the first record of each different answer are used. Some of the counters are used for different purposes under AUC and regular response mode. The names of the counter and their use is given below.

COUNTER	USE
N_XPOSED%	total number exposed to answer
N_SELECTORS%	total number of selectors for AUC-RIGHT : sum of 'soonness' of choice
T_SELECTORS	sum of the T-scores of the selectors for AUC-RIGHT : sum of squares of 'soonness' of choice
T2_SELECTORS	sum of the T-scores squared for AUC-RIGHT : sum of T-score * 'soonness' of choice
T_XPOSED	sum of the T-scores of those exposed to the choice (only used under AUC)
T2_XPOSED	sum of the T-scores squared of those exposed to the choice (only used under AUC)

5.6 POINTER FILE

Corresponding to each test bank is a pointer file. Records in this file consist of numbers which indicate the record number of the question header of each question in the test bank. Record 1 contains the number of questions in the entire test bank. Record 2 is a pointer to the header of question 1, record 3 points to question 2, etc. This file has a code number three more than the code of the corresponding test bank. In other words the pointer file for the test called RELATIVE for professor named

EINSTEIN, and for testbank 1 would be named 'RELATIVE4.EIN' (the corresponding test bank would be called 'RELATIVE1.EIN').

This file is formed and updated by the subroutine GET_QRCDS. This is done automatically after a test is entered or reviewed. This is a sequential file that has one numeric field.

5.7 OPTION FILE

The options are kept in a file with the same base as the testbank and pointer file, the same file type and the letter 'O' instead of a number at the end of the testname. For the previously given example (see above) the name of the option file would be 'RELATIVEO.EIN'.

This is a sequential file with three character fields. INFO$ contains the name of the option while VALUE$ contains the particular value for that option. LOCK$ is a single character which indicates whether or not the AUC/ONE option can be changed. Once a test has been administered under AUC or under regular response mode and the counters updated, the option cannot be changed until the counters are all set to zero. This field is set by the subprograms ENTER_OPTS, UPDATE_OPTIONS, and ZERO_COUNTERS. When the options are needed, the option file is opened and read until EOF is encountered. It only has nine records so that this is not a long process.

5.8 EXPOSURE FILE

The tags of all items exposed to an examinee are kept in the exposure file. This file has a code of 'X' ('RELATIVEX.EIN'). Each record contains one tag and a tag of zero is entered to indicate the end of a single examinee.

The exposure file is created by TAKE_TEST and used by UPDATE_COUNTERS. The professor must eliminate this file by choosing the option to 'delete the latest set of statistics' in the REPORTER program.

5.9 ANSWER FILE

The selections made by an examinee are kept in the answer file. This file has a code of 'A' ('RELATIVEA.EIN'). Each record represents the selections of one student. Each record has five fields. SNAME$ contains the student's name, first name first. SID$

contains the student's ID number (or letters). SCORE contains the raw score on the test, while TSCORE contains the student's T-score. ANS_ARR%(300) contains the tags of all selections made by the examinee. ANS_ARR%(0) contains the number of questions on the test. TSCORE is filled in when GRADE_REPORT is run. All other fields are entered at the time the student takes the test.

The answer file is created by TAKE_TEST and used by GRADE_REPORT and UPDATE_COUNTERS. The professor must eliminate this file by choosing the option to 'delete the latest set of statistics' in the REPORTER program. This is a fairly large file.

5.10 HARDCOPY FILES

All hardcopies are written to a sequential file called : TESTNAME$ + ".H" + NT$ ("1"<=NT$<="99"). The last two digits correspond to which of a number of different tests that file contains. Those files are printed and then deleted. The file name of the thirteenth test produced by HARD_COPY for the test RELATIVE would be 'RELATIVE.H13'.

5.11 HELP FILES

Each help file covers a single topic. They are all terminal format files and were entered with use of an editor. Each record is one field of 64 characters. The help routine forms the filename by using the parameters passed to it, and then prints out the contents of the file in sixteen lines per screen. The help files are kept in a subdirectory called [.TESTING.HELP]. This subdirectory is relative to the professors subdirectory [.TESTING]. The location of the help files is set in the subprogram HELP, contained in TESTSUBS1. In order to change the location of these files, simply make the change in the file name given at the begining of that subroutine.

5.12 SUBROUTINES

There are six files that contain external subroutines and one that contains functions. All are BASIC files except CLEARUP.FOR, a FORTRAN subroutine that clears the screen and raises the cursor on a VT-100, and DCLCOM.FOR, a FORTRAN subroutine that executes DCL (Digital Command Language) commands. NOTE : DCLCOM produces a log file in the home directory of the user. It deletes all but the latest version of these log files. This version, can either be left as is, deleted by hand, or deleted in a logout procedure.

The contents of each file along with the specific functions of the subroutines are listed below.

5.12.1 TESTSUBS1.BAS

TESTSUBS1 contains subroutines used by all parts of the testing program.

SUBROUTINE	PURPOSE
LCASE	changes all letters to lower case
UCASE	changes all letters to upper case
LEADELIM	eliminates leading blanks
BACKELIM	eliminates trailing blanks
SYSERROR	records system errors in the file TESTERR.ERR
MIXEMUPS1	randomly rearranges the elements of a one dimensional string array
MIXEMUPS2	randomly rearranges the elements of a two dimensional string array
MIXEMUPI2	randomly rearranges the elements of a two dimensional integer array
FILECOPY	copies one file to another
GET_QRCDS	fills the pointer file with the record numbers of the question headers for a particular test bank
ENCODE_FN	strips off leading characters of a string and makes sure the string is a valid file name
ENCODE	strips off leading N characters of a string
DECODE_TAG	decodes the answer tag into bank, question, and answer number
REM_QUO	finds the remainder and quotient of a number and its divisor
MARK_CHOICE	marks an answer tag as being chosen
CHECK_INPUT	asks the user to approve or change an input item
WARNING	prints a warning message about number of lines used
FIND_TEST	determines if a given file and/or test exists
DELAY	causes a delay equal to 0.1 sec * parameter
FORM_TESTNAME	gets the testname from the user
FORM_NUMBANKS	determines the number of test banks for a given test
UPDATE_HEADER	updates bank or question header in a test bank
HELP	prints out help messages
SWTSORT	switches names to last name first and sorts them

5.12.2 TESTSUBS2.BAS

TESTSUBS2 contains subroutines to assist in entering and revising a test.

SUBROUTINE	PURPOSE
REVIEW_TEST	used to review and revise an existing test
ENTER_TEST	used to enter a new test

5.12.3 TESTSUBS3.BAS

TESTSUBS3 contains subroutines that are used enter and revise the testing options.

SUBROUTINE	PURPOSE
ENTER_OPTS	enters new options for a test
USE_OLDOPTS	updates or leaves unchanged the old options
REVIEW_OPTS	reviews and revises old testing options

5.12.4 TESTSUBS4.BAS

TESTSUBS4 contains subroutines that help in forming and taking a test

SUBROUTINE	PURPOSE
GET_OPTS	gets the options from an option file for a specific test
WRITE_OPTS	prints out the options for a given test
FORM_TEST	creates an individualized test
SELECT_ANSWERS	randomly selects answers to be used on a test
COMMENTS	takes in anonymous student comments
TAKE_TEST	administers test to the student
STORE_ANS	stores the answers selected by the student
CHECK_ANS	gets confirmation on the student's choice
CHECK_FEEDBACK	supplies feedback if that option is chosen
SCORE_TEST	scores the test
SHOW_SCORE	displays the student's score at the end of a test

5.12.5 TESTSUBS5.BAS

TESTSUBS5 contains subroutines that help in performing the statistical analysis of a test.

SUBROUTINE	PURPOSE
ZERO_COUNTERS	zeros out all counters
UPDATE_COUNTERS	updates the counters with the latest statistics
DELETE_STATS	deletes the latest statistics
GRADE_REPORT	forms the grade report
ITEM_REPORT	forms the grade analysis
FIND_ADDR	finds the record number of the question that corresponds to the appropriate answer

5.12.6 TESTSUBSF.BAS

TESTSUBSF contains several external functions.

FUNCTION	PURPOSE
VALSTRING(X$)	returns the numeric value of a string variable
RANDNUM(UB,LB)	returns a random integer between UB and LB
FINDNXT(IP$,COL)	finds the next complete symbol in IP$ - the symbols are seperated by a colon.
FORM_TAG(B,Q,A)	forms the answer tag number 100101<=TAG NUMBER<=31009 the first digit represents the bank number, the next three digits the question number, the last two digits, the answer number
MODULO(N,D)	finds the value of N mod D

5.13 INSTALLATION AND EXECUTION OF THE PROGRAM

5.13.1 Placement Of The Files

Source and object files need not be kept on-line. Executable images should be kept in a subdirectory of the particular account in which they will be used. The subdirectory should be called [.TESTING].

Help files should be kept in a subdirectory called [.TESTING.HELP].

5.13.2 Compiling And Linking

If you do not have an executable image and need to create it follow the steps below. You will be creating three executable images.

i) Compile all BASIC and FORTRAN programs and subroutines

```
BASIC TESTENTRY
BASIC REPORTER
BASIC TAKETEST
BASIC TESTSUBS1
BASIC TESTSUBS2
BASIC TESTSUBS3
BASIC TESTSUBS4
BASIC TESTSUBS5
BASIC TESTSUBSF
FORTRAN CLEARUP
FORTRAN DCLCOM
```

ii) link the programs in the following manner

```
LINK TESTENTRY,TESTSUBS1,TESTSUBS2,TESTSUBS3,
     TESTSUBSF,CLEARUP,DCLCOM

LINK TAKETEST,TESTSUBS1,TESTSUBS4,TESTSUBSF,
     CLEARUP

LINK REPORTER,TESTSUBS1,TESTSUBS4,TESTSUBS5,
     TESTSUBSF,CLEARUP,DCLCOM
```

iii) you will now have 3 executable images :

```
TESTENTRY.EXE, TAKETEST.EXE, REPORTER.EXE
```

CHAPTER 6

USER'S MANUAL

This program can be used to create and implement an interactive testing package that has:

i) easy test entry and test generation

ii) randomized selection of questions

iii) on-line testing and grading

iv) statistical analysis of results

Test entry has been made as simple as possible with help given as needed so that an experienced user of the system will receive only a minimum number of prompts. To execute the first module, simply type 'RUN TESTENTRY'. This package allows a new test to be entered; an old test to be reviewed and possibly revised; a new set of testing options created or an old set of options used for a different test, either revised or unchanged.

The package is, as much as possible, self-prompting with help given as needed. You are first asked for your name and then addressed by that name along with the title 'professor', for the remainder of the program.

The first choice is which section of the package you care . to use. Selecting a choice invokes a subroutine to execute the action. The first list of choices is :

i) enter a new test
ii) review and possibly revise an existing test
iii) work on the testing options
iv) get help on the current topic
v) stop execution of the program

6. 1 ENTER A NEW TEST

6. 1. 1 Testname

You will first be asked to enter a test name. This name can be any set of less than nine letters and numbers, that begins with a letter. All of the subsequent information is stored in a file that is keyed to this name, your name, and a special tag that indicates the purpose of that particular file.

6.1.2 Password

You are then asked to enter a password. This can be any set of up to 20 characters. There is also the option of using no password. The password provides a level of security for entry into the test file, as well as to the program that generates the student's statistics.

When you choose a password it will be printed on the screen and you will be asked to check it. This is the only time it will appear on the screen. At other times when you are asked to check your password, it will not print on the screen. Once a password is chosen there is no way to change it, nor to review it by use of the program. Therefore one should exercise caution in choosing a password and make sure to rememeber it.

Each test can have a different password. It is strongly suggested that you use a password.

6.1.3 Question Types

You can then choose the type of question that you want to enter. There are two general types of question. These are multiple choice and keyed response.

6.1.3.a Multiple Choice Questions -

A multiple choice type question is one where the student is presented with a question stem and two or more possible responses. This program is limited to four prssible responses.

When entering this type of question you are first asked for the question stem. This can be one or several lines long. The question will be presented in the same manner in which it is entered. If more than twelve lines are entered, a warning is sent indicating that the question and the answers may take up more space than can be displayed on one screen. You can choose to ignore this warning. The screen holds 22 lines at a time and there is no way to see something that has gone off the top of the screen. When the test is administered, the student will be given a warning after twenty lines have been displayed. He will be told to remember what has been presented as it may soon go off the screen. The program waits for the student to press 'RETURN' before continuing putting up the questions or answers. Information will be better retained a screen at a time. Therefore, if you need to have very long questions and answers, it may be best to administer the test on paper. You can get printed copies of the test to be administered off-line.

You enter the question line by line, and terminate it by entering a blank line. You are then asked to check the question. You can change one line at a time, add additional lines, delete one line, or delete or rewrite the entire question. After accepting the question, you will be asked to enter a correct answer.

Correct answers are also entered line by line and you are given an opportunity to correct each answer. From one to forty-nine correct answers can be entered. When the test is formed, one of these will be randomly selected.

Incorrect answers are entered in a similar manner , and from one to forty-nine incorrect answers can be entered. When the test is formed, three, two or one incorrect answers will be selected. If there are three or more incorrect answers, then three will be used.

If more answers are entered for a particular question, then the test will have more variety. You do not want to enter selections just to fill space, but if even four incorrect responses are entered, different students will see a somewhat different test.

After entering one question, you will have the option of entering another question of the same type, changing question types, changing test banks, or stop entering question for the test.

NOTE : Changing an item may make the item statistics invalid, since some of the counters will be from the unchanged form of the item. You can delete the particular item and then re-enter it.

6.1.3.b Keyed Response Question -

For this type of question the student will key in a response which must exactly match one of a set of correct answers (as given by the professor). Spelling, capitalization, spacing, and punctuation are all considered in determining if the answer is correct. Therefore, when entering correct answers, try to include as many correct forms as possible. Entering a keyed response question is similar to entering a multiple choice question. The question can be one or more lines.

The package will not do any partial word recognition but will allow you to enter up to forty-nine correct answers. When the student enters an answer, it is checked against all entries in the bank and if it matches any one of the correct answer it is marked correct. It must match a correct answer exactly. Correct answers can only be one line long. There can be from one to 49 correct answers for each question. No incorrect answers are entered for this type of question. After you have finished entering the correct answers, you are presented with the same options that you get after finishing entering the incorrect answers for a multiple choice question.

6.1.4 Exit Menu

After entering each complete question you will have the following options.

i) enter another question

ii) change question type

iii) stop entering questions in this bank

iv) stop entering question for the test

v) get help

6.1.5 Test Banks

The package allows one, two, or three test banks with each bank containing from one to 100 questions. Typically the different banks will contain questions on material from different subject areas. A given number of question on the test will be selected from each bank. Questions from the same bank will be grouped together on the test.

You can review and revise questions from each test bank, however you cannot move questions from one test bank to another. Entering questions in the second and third test banks is done the same as it is for the first.

6.2 REVIEWING AND REVISING A TEST

If you are working on a test that you have entered previously, you will choose the option of review and revise an existing test. You will be asked for your name, the testname, and a password. The test must exist, and you must have entered your name and password correctly in order to continue with the review. NOTE : Your password will not print on the screen when you enter it and you will only get one chance to enter it. Once all of this information has been checked, you will be presented with the following options.

i) review the current question or answer
ii) skip the current question or answer
iii) insert another correct or incorrect answer
iv) add another question to the test bank
v) go to next question
vi) go to next test bank
vii) stop the review of the test
viii) get help

6.2.1 Review The Current Question Or Answer

If you choose this option you will be given a chance to look at the question stem, correct answer, or incorrect answer. You will view and be able to change each answer individually. You also can leave them unchanged. If you are working on an answer, you must view all the correct or incorrect answers.

For each question or answer, you can change one line at a time, add additional lines, delete one line, or delete or rewrite the entire question or answer. If you delete a question, the corresponding answers are automatically deleted. NOTE : When you change the number of lines in a question or answer, the entire file will be rewritten. Therefore this is not a fast process. Simply changing several lines without changing the size of a particular question or answer can be done quickly.

6.2.2 Skip The Current Question Or Answer

Choosing this option allows you to not review the current question, the correct answers, or the incorrect answers and move on to the next type. In other words, if you are working on a question stem and want to change only the correct answers use this option once. Once you skip one type (question stem, correct answers, incorrect answers), you will be given the same options for the next type.

6.2.3 Insert Another Correct Or Incorrect Answer

This option allows you to add an additional correct or incorrect answer for the particular question on which you are working. This involves rewriting the entire file, so it is not a fast process.

6.2.4 Add Another Question To The Test Bank

This option allows you to add an additional question to the test bank on which you are working. Question stems, correct and if applicable incorrect answers will be entered exactly as they are done when entering a test. You can enter as many questions as you choose at this time. When you are finished entering additional questions you will also be finished with the review of the test. All additional questions are added to the end of the test bank.

6.2.5 Go To Next Question

Choosing this option is a method of skipping through the test bank in order to review only certain questions. This is a fairly fast procedure, however you will need either to review the question stem or have a copy of the test bank in order to know on which question you are working. Questions are reveiwed sequentially from the begining of the test bank.

6.2.6 Go To Next Test Bank

This option stops the review of the current test bank and proceeds to the next test bank if it exists.

6.2.7 Adding A Test Bank

After you have originally entered a test, you can add an additonal test bank (up to 3) by taking the option 'go to next test bank' from the review test program.

When you choose this option. if there are no more test banks and you have nou used all three, you will be asked if you want to add a new test bank. If you answer 'YES' then you will proceed to the next test bank. You can enter up to 100 questions in the new test bank. This is done exactly as if you were entering a new test.

6.2.8 Stop The Review Of The Test / Help

These options do just what they say they do.

6.3 TESTING OPTIONS

All testing options will be placed in a file that is keyed to a professor's name and test name. When entering a second test, this file can be copied, updated, or reentered so that the second test can be taken with the same, somewhat revised, or completely different options respectively. The different testing and scoring options are discussed below. As far as is possible, the options are independent of one another. However, if AUC is chosen then feedback is automatically provided. Therefore the feedback option is not used.

6.3.1 Multiple Choice Questions

These questions can be answered under regular response mode (one choice per question) or under answer until correct (AUC). AUC means that the student will continue choosing questions until the right answer is chosen. If AUC is chosen, only multiple choice type questions with at least 3 incorrect answers will be used. For each question, 1 correct and 3 incorrect responses will be provided.

If regular response mode is chosen then multiple choice as well as keyed response type questions can be used on a test. For multiple choice questions, if there are 3 or more incorrect answers, then 1 correct and 3 incorrect answers will be used.

NOTE : Once this option is chosen, some tests administered using it, and the counters updated once, you cannot change the option until you have zeroed out the counters. Different item statistics are kept for regular response mode and AUC.

6.3.2 Feedback

If the feedback option is chosen it will consist solely of indicating the correct response, not of retesting nor giving extended explanations. The feedback option will be the same for all applicable questions (multiple choice and keyed response) on one test. NOTE : This option is only selected for regular response mode. If you are using AUC then feedback is automatic.

6.3.3 Number Of Questions

You will first be asked for the number of questions that you want to include on the entire test. You will then have the choice of using an equal number of question from each test bank, or selecting a specific number of questions from each test bank. If you are using only one test bank, then you will only be asked for the number of questions on the test.

No checking is done at this point to determine if enough questions exist in the test banks. When the test is created, the numbers are adjusted if there is any discrepency between what you chose and what actually exists.

6.4 SCORING

For regular response mode, all questions will receive one point for a correct response and no points for an incorrect response. This applies to both keyed respone and multiple choice questions.

Scoring on AUC will be based on the number of responses given relative to the total number of choices available. The points received will be equal to the number of choices not chosen. From 0 to 3 points will be awarded for each question.

6.5 REPORTER PROGRAM

Grade and item reports are prepared by running the program REPORTER. This program is protected by the same password that was used in entering the test. NOTE : Your password will not print on the screen when you enter it and you will only get one chance to enter it. After you have entered your name, the testname , and your password, you will be presented with the following options.

i) print copies of the test

ii) produce grade report

iii) update the item counters

iv) delete latest set of statistics

v) produce item analysis

vi) zero out item counters

vii) get help

viii) stop execution of the program

6.5.1 Print Copies Of The Test

This option allows you to print copies of the test. You will have the option of producing from zero to 99 different tests. Choosing to produce zero copies allows you to exit the program. You can then get from zero to 99 copies of each of the tests you produced. You must get the same number of copies of each of the tests. You can run this program more than once. Choosing to print zero copies will keep the tests in your account. The filename of the test is the TESTNAME you have chosen. The file type, is an 'H' followed by a number from 1 to 99 (corresponding to the number of tests that you produce).

The copies will be printed on regular computer paper and are deleted after they are printed. The output should appear in your regular output box. The last page of the test will be an answer key.

NOTE : When administering the test by paper, there is no automatic scoring nor statistical analysis. Therefore the item analysis will not include the results of those tests administered by paper. Statistics can be added by hand, but this is a slow process.

6.5.2 Produce Grade Report

This will produce a grade report on the last set of students to take the test. All students who have taken the test since you last deleted the statistics will be included in this report. In this report, you will get each student's name, ID, raw score, percentage, T-score, and the mean and standard deviation for the test. This program must be run before you update the item counters. It produces the T-scores based on the performance of the students in the report.

This is a fairly short report and does not take too long to produce.

NOTE : If the same student takes the test more than once, the last attempt will appear first on the grade report.

6.5.3 Update The Item Counters

This option will update the item counters. You should do this after a set of students have taken a test and you have produced a grade report for those students. Do not do this more than once for the same set of students.

6.5.4 Delete Latest Set Of Statistics

This will eliminate the results from the latest set of students to take the test. This should be done AFTER producing the grade report and updating the counters. You may produce the grade report and delete the statistics without updating the counters. Do this if you do not want to include that set of students in the item analysis.

6.5.5 Produce Item Analysis

The item analysis should be run after a substantial number of students have taken the test and the counters have been updated. With fewer than 100 total examinees, the item analysis will not be very meaningful.

The results of this report are explained in chapter 4 (TESTING AND MEASUREMENT).

NOTE : This report takes a long time to produce and is quite extensive. It contains information on every answer in the test bank.

6.5.6 Zero Out Item Counters

This program should be used only after you are finished gathering a complete set of statistics on a set of items. You may want to do this in order to change the mode of asking a question (regular response to AUC or visa versa). You also will want to do this if you copy a test bank from one test to another.

6.5.7 Get Help / Stop Execution Of The Program

These options do just what they say they do.

6.6 TAKETEST

In order for a student to take a test, he needs to go to the appropriate terminal, logon to the appropriate account, and type 'RUN TAKETEST' or some symbolic equivalent.

The student will be given instructions about the procedure for taking the test. If you are using the AUC option, you might want to explain how it works before administering the test.

For the entire process, the most important point to emphasize is that the students will be given ample opportunity to check their answers, so that they will not be penalized for typing mistakes. Computer, as well as testing anxiety, may be the biggest obstacles for students to overcome. After becoming familiar with the program, the student should no longer have any fear of the computer.

6.7 SECURITY

• The program is intended to be administered under supervision. All testing should be done from a single account.

Grade and item reports are deleted after they have been viewed and/or printed. The actual results of the test are in a form which is illegible without knowledge of how the program is structured. Nevertheless, all files should be protected against deletion.

All statistical information and test results can be written to a subdirectory in the professor's account. The image can be installed as a 'privileged image' and given WRITE access to that directory. Students in a certain 'group' can be given execute privileges to that image.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

Computer-assisted test construction and administration is a field that holds much promise. In the review of the literature, item banking, item generation, item selection, interactive and adaptive testing, and item types were discussed. For each area, there are a number of different methods of using a computer to aid in the production and administration of a test.

For the testing program produced for this paper, limitations were put on those features included. When given a choice between extra features and simplicity, the latter choice was taken. Some users may find the system lacking in one feature or another. There is no limit to what could be added to the program, and it was felt that additional features would not significantly add to the utility of the program.

Those features included provide the user with a complete interactive testing program, with detailed item analysis. The type and size of tests used is only limited by the desires of the users and the computer storage available; not by any limitations of the testing program. While each test is limited in the number of questions included, there is no limit on the number of tests that can be created nor on the size of any particular question or group of questions.

The testing program provides extensive item analysis and the capabilities of modifying, adding, or deleting any particular item (question stem, correct answer, incorrect answer). The program provides random selection of questions and answers from among those entered. Therefore a variety of tests can be formed when only a few extra choices are entered. This permits relatively easy test construction.

Item analysis statistics for regular response mode provide a measure of the difficulty of the question and of each answer, and of the type of students choosing a particular response. This allows the test creator to determine items which are either too hard, too easy, or are poor measures of ability. Those items which are not satisfactory can be eliminated or changed.

The AUC (Answer Until Correct) feature may be of special interest to users of the system. This method of testing is easily implemented on-line and is not prone to some scoring irregularities that could occur when the same method is used off-line. The item statistics for AUC are quite different from those used under regular response mode but they provide similar measures of item validity. This program provides a method for further study of testing under AUC.

The aim of computer-assisted test construction and administration shoud be to provide better as well as more easily prepared and administered tests. Better in that the test truly measures what it was designed to measure. This is aided by keeping and using well designed item statistics. The test should be more easily prepared and administered in that computer-assisted tests need not be rewritten nor retyped each time they are used. A different test can be produced each time with very little extra effort. Initial development of a test will take some time, and the test does need to be revised after it has been used. However, the end product will be a better constructed, easily administered test.

A final consideration in any computer-assisted testing package is that it must be used. The users will often be unfamiliar with the computer. Therefore any testing package must be user-friendly, and easy to use.

The testing program written for this paper, provides a means of achieving all of the above mentioned goals for computer-assisted test construction and administration.

CHAPTER 8

REFERENCES

[ALLEM79] Allen, M. J. and Yen, W. M. Intorduction to Measurement Theory, Monterey, California : Brooks/Cole Publishing, 1979.

[BORKA80] Bork, A. "Preparing On-Line Quizzes", ACM SIGCUE BULLETIN, 14, 4, 2-16, 1980.

[CALHW78] Calhoun, W. F. and Frary, R. B. "Computer-Supported Individualized Testing Within Reach", EDUCATIONAL TECHNOLOGY, 18, 1, 45-48, 1978.

[COLLR79] Collins, R. W. and Duff, S. J. "Computer-Assisted Test Construction via Automatic Program Generation: Using PROBGEN II to Create Individualized Exams and Problem Sets", PROCEEDINGS OF NATIONAL EDUCATIONAL COMPUTING CONFERENCE, 114-127, 1979.

[DENES78] Denenberg, S. A. "A Personal Evaluation of the PLATO System", ACM SIGCUE BULLETIN, 12, 2, 3-10, 1978.

[FRARR78] Frary, R. B., Ed. "Guessing on Multiple-Choice Tests", TEACHING AND LEARNING - TESTING MEMO 1, Learning Resources Center, Viginia Tech, Blacksburg, Virginia, November, 1978.

[FRARR79A] Frary, R. B., Ed. "Assigning Letter Grades to Test Scores", TEACHING AND LEARNING - TESTING MEMO 7, Learning Resources Center, Viginia Tech, Blacksburg, Virginia, May, 1979.

[FRARR79B] Frary,R.B., Ed. "How Difficult Should a Test Be", TEACHING AND LEARNING - TESTING MEMO 2, Learning Resources Center, Viginia Tech, Blacksburg, Virginia, February, 1979.

[FRARR79C] Frary,R.B., Ed. "What Kind of Grades Should Be Averaged?", TEACHING AND LEARNING - TESTING MEMO 6, Learning Resources Center, Viginia Tech, Blacksburg, Virginia, May, 1979.

[FRARR80] Frary,R.B. "Expected Multiple-Choice Test Item Scores Under Ordinal Response Modes", Paper presented at the annual meeting of the American Educational Research Association, Boston, 1980.

[FRARR82] Frary,R.B., Notes on item analysis.

[GARRH80] Garraway,H. "Microcomputer-Assisted Study and Testing System", PROCEEDINGS OF NATIONAL EDUCATIONAL COMPUTING CONFERENCE/2, 200-204, 1980.

[GARSJ80] Garson,J.W. "The Case Against Multiple Choice", THE COMPUTER TEACHER, 7, 4, 29-34, 1980.

[HISCM80] Hiscox,M.D. and Brezinski,E.J "A Guide to Item Banking in Education", paper presented at the June 1980 National Assessment Conference, 1980.

[LIPPG75] Lippey,G. "Computerized Test Construction", ACM SIGCUE BULLETIN, 9, 1, 13-14, 1975.

[SKAKE79] Skakun,E.N. and Petruk,M.W. "Computerized Test Production in the Medical Field", PROCEEDINGS OF THE ANNUAL CONVENTION OF THE ASSOCIATION FOR THE DEVELOPMENT OF COMPUTER-BASED INSTRUCTIONAL SYSTEMS, II, 705-710, 1979.

[SORLW79] Sorlie,W.E., Essex,D. and Shatzer,J. "A Computer-Based Student Appraisal System: Automated from Sign-On to Item Analysis", PROCEEDINGS OF THE ANNUAL CONVENTION OF THE ASSOCIATION FOR THE DEVELOPMENT OF COMPUTER-BASED INSTRUCTIONAL SYSTEMS, II, 652-662, 1979.

[STOCM80] Stocking, M. L. and Swanson, L. "Computerized Adaptive Testing", THE COMPUTER TEACHER, 7, 4, 12-13, 1980.

[SUBRR75] Subramanian, S. R. "Overview of Computer-Assisted Test Construction", ACM SIGCUE BULLETIN, 9, 1, 12-13, 1975.

[SWANL80] Swanson, L., Stocking, M. and Kreitzberg, C. "Computers Open the Door to Tailoring Tests to the Individual", THE COMPUTER TEACHER, 7, 4, 15, 1980.

[THOMS81] Thompson, S. B., Richard, L. H. and Haddett, M. J. "Empirical Effects of Partial Credit Grading, Non-uniform Item Weighting and Item Type on Math Achievement Test Validity", Paper presented at the annual meeting of the American Educational Research Association, Los Angeles, 1981.

[WALLS76] "CATC: Computer Assisted Test Construction Conference", ACM SIGCUE BULLETIN, 10, 1, 14-17, 1976.

CHAPTER 9

APPENDIX - SUMMARY OF THE USER'S MANUAL

9.1 TESTENTRY

This program is used to enter a new test, revise an existing test, or to work on testing options.

MAIN MENU FOR TEST ENTRY

-enter a new test
Used to create a new test.

-review and possibly revise an existing test
Used to change, view, delete from, and/or add to an existing test. The test must exist.

-work on the testing options
Used to enter, view and/or change testing options.

QUESTION TYPES

-MC

A multiple choice type question is one where the student is presented with a question stem and from two to four possible responses. The student will then chose from among the responses.

-KR

A keyed response question is one in which the student will key in a response which must exactly match one of a set of correct answers. The question can be one or more lines. Correct answers can only be one line long. No incorrect answers are entered. Student answers must match a correct answer exactly.

EXIT FROM TESTENTRY

-enter another question

Add another question of the same type to the test bank. Each test bank can have up to 99 questions.

-change question type

Choose from multiple choice and keyed response type questions.

-stop entering questions in this bank

Move on to the next test bank. You are permitted a total of 3 test banks for the entire test.

-stop entering question for the test

You will move on to entering the testing options

REVIEW TEST MENU

-review the current question or answer

You will be given a chance to look at the question stem, correct answer, or incorrect answer. If you are working on an answer, you must view all the correct or incorrect answers. For each question or answer, you can leave it unchanged, change one line at a time, add additional lines, delete one line, or delete or rewrite the entire question or answer. If you delete a question, the answers are automatically deleted.

-skip the current question or answer

You will not review the current question, the correct answers, or the incorrect answers, but will move on to the next type.

-insert another correct or incorrect answer

This option allows you to add an additional correct or incorrect answer for the particular question on which you are working. This is not a fast process.

-add another question to the test bank

This option allows you to add an additional question to the end of the current test bank. Question and answers will be entered exactly as they are done when entering a test. You can enter as many questions as you choose at this time. When you are finished entering additional questions you will also be finished with the review of this test bank.

-go to next question

This option is a method of skipping sequentially through the test bank in order to review only certain questions.

-go to next test bank

This option stops the review of the current test bank and proceeds to the next test bank if it exists.

-stop the review of the test

Stops the review of all test banks.

MAIN MENU FOR ENTERING OPTIONS

-enter new options
Use this to produce new options.

- use old options, revised or unchanged
Use the existing options for an existing test.

MULTIPLE CHOICE QUESTIONS OPTION

ONE or regular response mode
The student can choose only one answer from the set provided. (1 correct and up to 3 incorrect) If regular response mode is chosen then multiple choice as well as keyed response type questions can be used on a test. All questions under reqular respone mode will receive one point for a correct response and no points for an incorrect response. This applies to both keyed respone and multiple choice questions.

Answer Until Correct (AUC)
AUC means that the student will continue choosing answers until the right answer is chosen. If AUC is chosen, only multiple choice type questions with at least 3 incorrect answers will be used. All other question will be skipped. For each question, 1 correct and 3 incorrect responses will be provided as possible choices. The points received on an AUC question will be equal to the number of choices not chosen. From 0 to 3 points will be awarded for each question. NOTE : Once this option is chosen, some tests administered using it, and the counters updated once, you cannot change the option until you have zeroed out the counters.

FEEDBACK OPTION

When using the feedback option, the student is told whether or not his choice is correct. If it is incorrect then he will be given the correct response to that question. There is no provision for retesting nor for giving extended explanations. The feedback option will be the same for all questions on one test. NOTE : This option is only selected for regular response mode. If you are using AUC then feedback is automatic.

NUMBER OF QUESTIONS ON TEST

You are to enter the number of questions that you want to include on the entire test. You will then have the choice of using an equal number of question from each test bank, or selecting the number of questions from each bank.

If you choose to use an equal number of questions from each test bank, the the total number of questions on the test will be divided by the number of testbanks, and that number used on each testbank. If you choose not to take an equal number from each test bank, then you will select the number you want to use from each bank.

If you choose to use less questions than exist in the test banks, then a variety of tests will be created. You should not make tests too long. This option can be changed at any time, but should be the same for one class of students. No checking is done at this point to determine if enough questions exist in the test banks. When the test is created, if there is any discrepency between what you chose and what actually exists, then the numbers are adjusted.

9.2 REPORTER

This program is used to run the statistical analysis for the test. It is protected by a password.

MAIN MENU FOR REPORTER

-produce printed copies
This allows you to get printed (hard) copies of the test. You will have the choice of from 0-99 different tests and from 0-99 copies of each test.

-produce grade report
This will produce a grade report on the lastest set of students to take the test. You will get raw scores, percentages, T-score, and the mean and standard deviation for the test. This program must be run before you update the item counters.

-update the item counters
You should do this after a set of students have taken a test and you have produced a grade report.

-delete latest set of statistics
This will eliminate the results from the latest set of students to take the test. This should be done AFTER you have produced the grade report and updated the counters. You may produce the grade report and delete the statistics without updating the counters.

-produce item analysis
The item analysis should be run after several groups of students have taken the test and the counters have been updated.

-zero out item counters
This program should be used only after you have gathered a complete set of statistics on a set of items. You should do this in order to change the mode of asking a question. You also will want to do this if you copy a test bank from one test to another.

The two page vita has been removed from the scanned document. Page 1 of 2

The two page vita has been removed from the scanned document. Page 2 of 2

INTERACTIVE TESTING PROGRAM

by

Scott H. Stuart

(ABSTRACT)

This paper is a discussion of the research, design, and implementation of an interactvie testing program that was written in VAX-BASIC, and implemented on a VAX-11/780.

Computer-assisted test construction and administration are discussed. Free response test items and response selection test items are discussed and implemented as are test, student, and item analysis statistics.

The program allows the user to enter, change, delete, or add, question stems, correct answers, and incorrect answers. The program will randomly select questions and appropriate answers from the created test bank(s). It will either administer and score the test interactively, or produce a printed copy of the test along with an answer key. A statistical package allows the user to produce grade reports and item statistics.

The program allows a professor to develop, and administer high quality test.