DOCUMENT RESUME

ED 395 972 TM 025 110

AUTHOR	Bennett, Randy Elliot; And Others
TITLE	Free-Response and Multiple-Choice Items: Measures of the Same Ability?
INSTITUTION	Educational Testing Service, Princeton, N.J.
REPORT NO	ETS-RR-90-8
PUB DATE	Jun 90
NOTE	41p.
PUB TYPE	Reports - Evaluative/Feasibility (142)
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Ability; Advanced Placement; College Entrance Examinations; Factor Structure; Goodness of Fit; High Schools; *High School Students; *Measurement Techniques; *Multiple Choice Tests; Scoring; Student Placement; Test Construction; *Test Items; Test Use
IDENTIFIERS	*Advanced Placement Examinations (CEEB); Confirmatory Factor Analysis; *Free Response Test Items

ABSTRACT

This study examined the relationship of multiple-choice and free-response items contained on the College Board's Advanced Placement Computer Science (APCS) examination. Subjects were two samples of 1,000 randomly drawn from the population of 7,372 high school students taking the 1988 examination of the APCS "AB" form. Most were high school seniors, most were male, and most were white. Confirmatory factor analysis was used to test the fit of a two-factor model where each item format marked its own factor. Results showed a single-factor solution to fit the data best in each of the two random-half samples. This finding might be accounted for by several mechanisms, including overlap in the specific processes assessed by the multiple-choice and free-response items and the limited opportunity for skill differentiation afforded by the year-long APCS course. Appendix A contains an example of the scoring rubric, and Appendix B presents a sample correlation matrix. (Contains 2 figures, 6 tables, and 21 references.) (Author/SLD)

RR-90-8

RESEARCH REPORT

FREE-RESPONSE AND MULTIPLE-CHOICE ITEMS: MEASURES OF THE SAME ABILITY?

Randy Elliot Bennett
Donald A. Rock
Minhwei Wang

Educational Testing Service
Princeton, New Jersey
June 1990

Free-Response and Multiple-Choice Items: Measures of the Same Ability?

Randy Elliot Bennett

Donald A. Rock

and

Minhwei Wang

Educational Testing Service

Princeton, NJ

Abstract

This study examined the relationship of multiple-choice and free-response items contained on the College Board's Advanced Placement Computer Science (APCS) examination. Confirmatory factor analysis was used to test the fit of a two-factor model where each item format marked its own factor. Results showed a single-factor solution to fit the data best in each of two random-half samples. This finding might be accounted for by several mechanisms, including overlap in the specific processes assessed by the multiple-choice and free-response items, and the limited opportunity for skill differentiation afforded by the year-long APCS course.

Index Terms: constructed-response items, free-response items, open-ended items.

Free-Response and Multiple-Choice Items: Measures of the Same Ability?

Many questions can be raised about the potential differences between multiple-choice and free-response item formats. Some of these questions concern measurement characteristics, in particular differences in traits measured, predictive power in applied settings, reliability, and the interactions of these characteristics with such factors as race and gender. Other questions regard the operational implications of the types for large-scale programs, for example, timing, cost, and scoring complexity. Finally, there are issues of pedagogical value (J. R. Frederiksen & Collins, 1989; N. Frederiksen, 1984) and of face validity.

This paper is concerned with one particular measurement characteristic, the equivalence of the traits measured by the two item formats. The extent of equivalence is of particular interest because the two formats are often portrayed popularly and in the educational research community as not only measuring disparate cognitive constructs, but measuring ones of different value (Fiske, 1990; Nickerson, 1989). Particularly, multiple-choice tests are depicted as assessing simple factual recognition and free-response as evaluating higher-order thinking. Such potential differences are of serious concern, for among other things, they imply a mismatch between the highly-valued thinking skills schools are lately attempting to impart and the methods used for determining if those goals are being achieved.

In a recent review of the literature on format effects in achievement testing, Traub and MacRury (in press) concluded that the two formats did appear to measure somewhat different abilities but that the nature of these differences was unclear. N. Frederiksen (1984, 1990) has argued that part of this ambiguity is owed to comparisons in which the free-response questions are specifically constructed to differ from existing multiple-choice items only in response format. In such cases, the constructed-responses will measure the same limited skills as the multiple-choice items.

The present study was intended to assess the trait equivalence of multiple-choice and free-response items in computer science. This domain is particularly interesting because of the College Board's Advanced Placement Computer Science (APCS) Examination. The APCS examination contains both multiple-choice and free-response questions written to measure the same content, but with the latter intended to more deeply assess selected topics, such as programming methodology (College Board, 1989). The free-response items would appear to be more than simple adaptations of the multiple-choice questions and, therefore, largely free of the limitations that concerned N. Frederiksen (1984, 1990). Consequently, the APCS test should provide a reasonable opportunity for any real differences in the underlying traits measured by these formats to emerge.

Some indication of the extent of format differences associated with the APCS examination can be gained from a study by Bennett et al. (in press). This study used APCS multiple-

choice and free-response formats as construct validity criteria for an intermediary item type and, hence, indirectly examined the relationship of multiple-choice to the free-response format. The study found little support for the existence of trait differences between the formats. The present study directly tests this relationship and, in addition, uses larger, unselected examinee samples (as opposed to volunteers) and longer multiple-choice and free-response tests.

Method

Subjects

Subjects were two samples of 1000 randomly drawn from the population of 7,372 high school students taking the 1988 administration of the APCS "AB" examination. The majority of subjects identified themselves as seniors (69% in sample 1, 67% in sample 2), with most of the remainder indicating junior class status (26% and 28%, respectively). Students in both samples were overwhelmingly male (86% in sample 1 and 87% in sample 2) and most were white (70% and 69% respectively). The largest single minority group was of Asian/Pacific Islander descent (16% in sample 1, 17% in sample 2).

Instrument

The APCS "AB" examination is intended to assess mastery of topics covered in a college-level introductory course in computer science (College Board, 1989). It emphasizes programming methodology and procedural abstraction, algorithms, data structures, and data abstraction. Two sections containing 35 and 15 items, respectively, compose the multiple-choice portion.

Coefficient alpha estimates for number-right raw scores computed for the 50 multiple-choice items were .89 and .88 for samples 1 and 2, respectively.

The free-response portion is made up of two sections; three questions compose the first section and two items the second. Items require the student to write or design a program, subprogram, or data structure, and at times to analyze the efficiency of certain operations involved in the solution. Coefficient alpha for the sum of the five partial-credit free-response scores was .78 in sample 1 and .77 in sample 2.

The four test sections are administered on the same day and are separately timed. This timing arrangement allows an abridged version of the test--the "A" examination, consisting of the first multiple-choice and first free-response sections--to be administered separately to students taking only the first semester of the APCS course. For both versions, the multiple-choice sections precede the free-response ones. Examples of the two item types can be found in Figures 1 and 2.

Insert Figures 1 and 2 about here

Procedure

A two-factor model composed of multiple-choice and free-response factors was posed to test the relationship of the skills measured by the multiple-choice and free-response items. The first factor was marked by parcels of multiple-choice items. Five ten-item parcels were constructed by randomly assigning

questions stratified on the basis of test specification content area. The content areas were programming methodology (11 items), features of programming languages (15), data types and structures (7), algorithms (13), computer systems (3), and applications (1). Items were then shifted among parcels (but within content categories) so that the mean difficulty values for the parcels were similar (mean values on the 0-10 number-right raw-score scale ranged from 5.06 to 5.77 in sample 1 and 4.89 to 5.75 in sample 2). The second factor was indicated by five APCS free-response problems. Each item was scored on a ten-point scale according to an analytical scoring rubric (see Appendix A) applied by a single reader, with five different individuals usually scoring the five responses for any single examinee.

Table 1 depicts the factor pattern matrix for the hypothesized model. The asterisks indicate that a factor loading was to be estimated. Conversely, a "0" denotes that the indicator variable was constrained to have a zero loading on that factor. The maximum likelihood factor estimation procedure from EQS (Bentler, 1989) was used to estimate the unknown factor loadings (i.e., the asterisks) from the sample covariance matrix subject to the pattern of zero constraints and allowing the factors to be intercorrelated. (See Appendix B for the input matrices, which are presented in the correlational metric for ease of interpretation.)

Insert Table about 1 here

Because the distributions for some of the markers were non-normal (particularly for the free-responses), the factor pattern was also estimated using the EQS generalized least squares solution procedure. This procedure provides for asymptotic standard errors and overall goodness-of-fit tests that do not assume normality. These results, however, produced no substantive difference from those estimated using the maximum likelihood procedure and, consequently, it is the maximum likelihood results that are reported here.

The fit of the two-factor model was assessed by examining its factor intercorrelations and goodness-of-fit indicators, and by comparing the model's fit to two alternatives: (1) a null model in which no common factors were presumed to underlie the data (i.e., each of the ten markers was allowed to load only on its own factor) and (2) a general model in which all variables loaded on a single factor. These alternative models allowed the goodness-of-fit indices to be investigated as a function of factorial complexity, where changes in the indices suggest how much fit is lost by moving from more to less complex models.

In confirmatory factor analysis, universally accepted measures of fit do not exist (Marsh & Hocevar, 1985; Sobel & Bohrnstedt, 1985). Even though statistical tests are available (e.g., the chi square test), these tests are highly sensitive to sample size, and may permit trivially false models to be rejected with large samples and grossly false ones to be accepted in small samples (Bentler & Bonnett, 1980; Marsh, Balla, & McDonald, 1988). Because hypothesized models are best regarded as

approximations to reality, the models will always be false to some degree making the interpretive task one of determining how reasonable a given model is. This judgment is typically based on the simultaneous evaluation of several goodness-of-fit indicators.

In the present investigation, the following indicators were used:

Chi-square/degrees of freedom ratio. This index is based upon the overall chi-square goodness-of-fit test associated with each factor model. Ratios of 2.0 or lower are commonly taken as evidence of good fit, though some investigators have suggested accepting values of up to 5.0 (Marsh & Hocevar, 1985). This index's sensitivity to sample size would appear to require extending even this limit when large samples are employed (Marsh, Balla, & McDonald, 1988).

Nonnormed fit index (NNFI). The nonnormed fit index is an adaptation of the Tucker-Lewis index (Tucker & Lewis, 1973), which represents the reliability of the hypothesized solution. The NNFI assesses the fit of a model with reference to the baseline null model, scaling fit from equivalent to the null model to perfect fit (Loehlin, 1987). The index can occasionally fall outside the 0-1 range, with larger values indicating better fit.

Akaike information criterion. The Akaike information criterion (AIC) is an index of parsimony that takes into account both the statistical goodness of fit and the number of parameters

that have to be estimated to achieve that degree of fit (Bentler, 1989). For the AIC, the smaller the index the better the fit.

Hierarchical chi-square test. These tests help in determining which of two models that share a nested relationship has the better fit (Loehlin, 1987). The chi-square for this test is the difference between the separate chi-squares of the two models. The number of degrees of freedom is computed analogously.

Standardized residuals. Standardized residuals can be used both to judge fit and to locate the specific causes of a lack of fit. If the model is a good representation of the data, the residuals should be symmetric and centered around zero (Bentler, 1989). Standardized residuals can be interpreted in the metric of correlations among the observed variables. The average off-diagonal absolute standardized residual summarizes the average correlation among the markers that is left over after the hypothesized model has been fitted.

Results

Table 2 presents APCS means and standard deviations for the two study samples and for the population taking the 1988 APCS examination. (Scores in this and all other analyses are number-right raw score as opposed to the formula scores used in the APCS program.) As the table suggests, the samples appear to closely represent the APCS population.

Insert Table 2 about here

Table 3 presents the loadings for each variable as estimated from the two-factor model. In both samples, all loadings are significant ($p < .001$., z-range for sample 1 = 20.30 to 31.37; z-range for sample 2 = 19.38 to 30.42. Loadings for the multiple-choice factor are generally slightly higher than those for the free-response factor. This difference might be due to the lower reliability of the free-response items or to the fact that the multiple-choice indicators were constructed so as to be parallel, causing them to share more variance. Each free-response indicator, in contrast, deals with a different topic, thereby reducing the common variance and, hence, the loading of each on the common factor.

Insert Table 3 about here

Goodness-of-fit indices and standardized residuals suggest the extent to which the model is complex enough to account for the data. For the two-factor model, the chi square/degrees of freedom ratio was 3.68 in sample 1 and 3.18 in sample 2, possibly inadequate in smaller samples but quite reasonable for sample sizes of 1000. This judgment is supported by the NNFI which, at .98 in both samples, suggests that the two-factor model accounts for virtually all of the reliable variance among the markers. The average off-diagonal absolute standardized residuals (AODASR)--which indicate the average correlation among the markers left after the two-factor model is fitted--provide additional confirmation. The AODASRs were .02 for both samples;

compared with median observed correlations among the markers of .52 and .47 for samples 1 and 2 respectively, these values show little remaining covariation. Last, the standardized residuals themselves were closely centered around zero, falling between 0 and .1 in magnitude in both samples.

Factor intercorrelations suggest whether a simpler model might also account for the data. The disattenuated correlation between the factors was .97 in sample 1 and .93 in sample 2. Each correlation was tested for its difference from 1.00 via a t-test using the standard error of estimate generated by the factor model. The correlations in both samples were significantly different from unity (the 99% confidence intervals were .939 to .999 in sample 1 and .890 to .968 in sample 2). However, the magnitude of these differences is so small as to question whether a simpler model might capture the data almost as well.

Further insight on the need for the two-factor model is gained from comparing it to the alternatives (see Table 4). For both samples, no loss or a minimal loss in fit occurs in moving from the two- to the single-factor solutions, though substantial lack of fit occurs when the null model is reached. For example, the chi-square/degrees of freedom ratio changes by less than a point from the two-factor to the single-factor models, but increases by over a hundred points from the single-factor to the null solutions.

Insert Table 4 about here

Table 5 presents hierarchical chi-square tests for the competing models. In both samples, these tests indicate significant improvements in fit for the single-factor over the null model. Although the tests also show significant improvements for the two-factor over the single factor model, the practical value of these improvements must be strongly questioned given the trivial gains suggested by the other fit indices.

Insert Table 5 about here

Finally, relative fit also can be assessed by examining the distributions of the standardized residuals (not shown). The residuals in both samples were distributed virtually identically for the two- and single-factor models, falling between 0 and .1 in absolute value. Only one value, associated with the single-factor solution, fell outside this range. This value, at .14, constituted a trivial departure.

Table 6 shows the loadings for the single-factor solution. Again, all loadings are significant ($p < .001$; z-range for sample 1 = 20.10 to 31.24; z-range for sample 2 = 19.06 to 30.16). As for the two-factor solution, the loadings for the multiple-choice markers are slightly higher than those for the free-responses. The probable explanations are similar: higher reliability and smaller content differences across markers (being parallel, the multiple-choice markers share more variance and, consequently, play a bigger role in defining the common factor than do the free-response indicators).

Insert Table 6 about here

Discussion

This study examined the interrelationship of multiple-choice and free-response questions contained on the College Board's Advanced Placement Computer Science Examination. Results suggested that a single factor provided the most parsimonious fit.

As noted, N. Frederiksen (1984, 1990) has contended that such findings are generally associated with investigations in which free-response questions have been adapted from existing multiple-choice items, and hence measure the same limited skills as their counterparts. We have argued that the APCS examination is an interesting environment for evaluating the trait equivalence of these formats because the free-response items are developed to measure certain content more deeply than the multiple-choice questions. Though these free-response items do not represent the task complexity typical of real-world programming environments (or even some introductory college-level courses), it is difficult to characterize the items as trivial, factual recall questions.

Some speculations on the processes these free-response items measure might suggest what underlies the high relationship between performance on the two formats. Research on the development of programming competence suggests that successful programmers map problem specifications into a deep-structure,

goal and plan representation, where goals are the objectives to be achieved in a program and plans the stereotypical means (i.e., step-by-step procedures) for achieving those goals (Soloway & Ehrlich, 1984; Soloway & Iyengar, 1986). As such, a free-response problem of the type presented on the APCS exam would appear to require the student to decompose the specification into goals, formulate plans to achieve each goal, translate each plan to Pascal code, and then debug that code by mentally simulating its effects. Depending on the results of this mental simluation, the examinee may return to an earlier step in this process: the simulation may suggest errors in the decomposition, the plans, or the translation of plans into code.

Accepting for the moment that this is a reasonable approximation of the processes involved in responding to the APCS free-response questions, one hypothesis is that the multiple-choice items measure some of these same processes. Given their nature, it is difficult to imagine any single multiple-choice item capable of assessing much more than one of these processes. However, it is plausible that in combination, 50 such items might cover in some depth many of the processes tapped by the free-response questions.

Some indication of this hypothesis' plausibility can be gained from an informal classification of the multiple-choice items in relation to the processes presumably required by the free-response questions. From this categorization, it appears that about half of the multiple-choice section (25 items) requires direct operations on Pascal code, in particular,

mentally simulating the code to predict the result, to identify how it should be changed to achieve a desired outcome, to compare it with an alternative method for achieving that result, or to describe its function, among other things. Item number 13 in Figure 1 is typical of these questions. These items would logically appear to be closely tied to translating plans into code and to debugging.

An additional seven items call for knowledge about Pascal or more general programming conventions but do not require mental simulation. These items ask the examinee to identify the differences between common control structures (e.g., while and repeat-until), specify reasons for using value versus variable parameters, and recall the rules of Pascal to determine how given variables can be legitimately used. These items also would appear to be related to the coding process. Item #5 in Figure 1 is an example.

A third class of items appears more related to plan formulation than coding. These 13 items focus on general knowledge of algorithms and data structures: comparing the efficiency of two search algorithms, identifying common characteristics of stacks and queues, and comparing the appropriateness of alternative data structures to a given specification. Item #9 exemplifies this category.

Finally, five items seem to be targeted at general computing knowledge: identifying the most user-friendly interface, recognizing the definition of "top-down," and indicating the original purposes of Pascal. Item #1 is an example. These

questions would obviously appear to be less closely tied to the free-response items than the other questions would.

This informal analyis suggests that the overwhelming majority of multiple-choice items do overlap with the free-response questions in some of the processes called for. Additional mechanisms might contribute to the strong relationship between the two types. For one, the way the domain is structured and taught might have some bearing. The content of the AP Computer Science course is taught during a single year. Consequently, there is relatively little opportunity for differentiation, for students to develop strengths in particular subdomains or in processes that might be better measured by one or the other item type (e.g., coding vs. problem decomposition). Second, the item types might invoke different processes that are not well-captured by factor analytic methods. Factor analysis is driven by individual differences. If the level of skill in implementing a particular process is sufficiently low in relation to the examinees' abilities to execute it, there will be no variation among examinees in the process and factor analysis will not reveal any distinction between items that do and do not require it. Such an eventuality might have occurred with the more difficult free-response items, on which many examinees received low scores.

Some of these speculations might be resolved by posing and testing plausible process-oriented factor models. One possibility is to examine the relations among the free-response questions and the multiple-choice item classes defined above to

see if expectations about those relations are supported. A second approach would be to elaborate more completely a psychological model for responding to free-response questions, specifically construct multiple-choice items to tap each of the processes, and test the hypothesized relations to see if they can be empirically confirmed.

The present finding of format equivalence needs to be carefully delimited. One such delimitation is to the computer science domain. This point deserves special emphasis given Traub and MacRury's (in press) conclusion that the formats are not generally equivalent and given the specific demonstrations of this non-equivalence in such domains as divergent thinking (N. Frederiksen & Ward, 1978; Ward, N. Frederiksen, & Carlson, 1980).

A second delimitation is to the APCS population. As noted, this population might show a relatively uniform skill profile because of the brevity of the APCS course. Greater skill differentiation, and perhaps more discernable item type differences, might be evident for individuals with more experience (e.g., graduate students specializing in computer science).

Third, these results should be limited to the tasks presented. A fair number of the APCS multiple-choice items appear to require some of the higher-order skills commonly attributed to free-response questions. At the same time, the APCS free-response tasks, though arguably non-trivial, represent neither the length nor the complexity of real-world programming problems. Different results might occur with multiple-choice

items targeted more towards factual recognition or free-response questions requiring more extended or complicated productions.

A fourth limitation on generalizability is the method used to score the free-responses. One of the major attractions of these items is their potential for elucidating a rich response. Because of this richness, the same response might be scored simultaneously along several dimensions including responsiveness to the specification, efficiency, user-friendliness, and originality. The analytical scoring scheme used in APCS does not take full advantage of this richness, combining some of these dimensions in a single score and not considering others (e.g., originality). Better capturing the richness of these solutions might produce measures more distinct from multiple-choice.

A final, and perhaps most important, delimitation is to assessment purpose. There are good arguments to be made for the non-equivalence of the two formats for instructional diagnosis (Bennett, in press; Birenbaum & Tatsuoka, 1987). Free-response can provide a trace of the examinee's solution process that is not easily duplicated by multiple-choice. Such processes may reveal not only partial knowledge, but also different erroneous approaches to the problem given the same level of knowledge.

Also with respect to assessment purpose is that even with factor intercorrelations in the .90s, the factors theoretically can predict a third variable to dramatically different degrees. Consequently, there may be some prediction situations for which the item types might not be equivalent.

Last, the item types are probably not equivalent for some of the purposes of the APCS examination. This exam is primarily intended to assess course mastery associated with advanced placement or college-level credit. If the exam's only purpose was to determine course mastery, the multiple-choice format might be preferred solely for its efficiency. The examination is, however, intended to do more. The free-response section serves to make visible to teachers and students behaviors considered important to course mastery; without this visibility there is the danger that instruction might emphasize the tasks posed by the multiple-choice section to the exclusion of programming, one of the central components of computer science. In addition, the grading of the free responses serves important ends. For the annual grading, selected APCS teachers are brought together from all over the country for a one-week period. This event gives APCS teachers an opportunity to learn free-response standard-setting and grading techniques, share classroom experiences, and play an integral part in the examination process, thereby developing a sense of ownership in the AP program.

In sum, the evidence presented offers little support for the stereotype of multiple-choice and free-response formats as measuring substantially different constructs (i.e., trivial factual recognition vs. higher-order processes). All the same, there are often sound educational reasons for employing the less efficient format, as some large-scale testing programs, such as AP, have chosen to do.

References

Bennett, R. E. (In press). Intelligent assessment: Toward an integration of constructed-response testing, artificial intelligence, and model-based measurement. In N. Frederiksen, R. J. Mislevy, and I. Bejar (Eds), Test theory for a new generation of tests, Hillsdale, NJ: Lawrence Erlbaum Associates.

Bennett, R. E., Rock, D. A., Braun, H. I., Frye, D., Spohrer, J. C. & Soloway, E. (In press). The relationship of constrained free-response items to multiple-choice and open-ended items. Applied Psychological Measurement.

Bentler, P. M. (1989). EQS Structural equations program manual. Los Angeles: BMDP Statistical Software.

Bentler, P. M., Bonnett, D. G. (1980). Significance tests and goodness of fit in the analysis of covariance structures. Psychological Bulletin, 88, 588-606.

Birenbaum, M., & Tatsuoka, K. K. (1987). Open-ended versus multiple-choice response formats--It does make a difference for diagnostic purposes. Applied Psychological Measurement, 11, 385-395.

Fiske, E. B. (1990, January 31). But is the child learning? Schools trying new tests. The New York Times, pp. 1, B6.

The College Board. (1989). Advanced Placement course description: Computer science. New York: Author.

Frederiksen, J. R., & Collins, A. (1989). A systems approach to educational testing. Educational Researcher, 18(9), 27-32.

Frederiksen, N. (1984). The real test bias: Influences of testing on teaching and learning. American Psychologist, 39, 193-202.

Frederiksen, N. (1990). Introduction. In N. Frederiksen, R. Glaser, A. Lesgold, & M. G. Shafto, (Eds). Diagnostic monitoring of skill and knowledge acquisition. Hillsdale, NJ: Lawrence Erlbaum Associates.

Frederiksen, N., & Ward, W. C. (1978). Measures for the study of creativity in scientific problem solving. Applied Psychological Measurement, 2, 1-24.

Loehlin, J. C. (1987). Latent variable models. Hillsdale, NJ: Erlbaum.

Marsh, H. W., Balla, J. R., McDonald, R. P. (1988). Goodness-of-fit indexes in confirmatory factor analysis: The effect of sample size. Psychological Bulletin, 103, 391-410.

Marsh, H. W., & Hocevar, D. (1985). Application of confirmatory factor analysis to the study of self-concept: First and higher order factor models and their invariance across groups. Psychological Bulletin, 97, 562-582.

Nickerson, R. S. (1989). New directions in educational assessment. Educational Researcher, 18(9), 3-7.

Sobel, M. E., & Bohrnstedt, G. W. (1985). Use of null models in evaluating the fit of covariance structure models. In N. B. Tuma (Ed), Sociological Methodology. San Francisco: Jossey-Bass. pp 152-178.

Soloway, E., & Ehrlich, K. (1984). Empirical investigations of programming knowledge. IEEE Transactions on Software Engineering, 10, 595-609.

Soloway, E., & Iyengar, S. (Eds). (1986). Empirical studies of programmers. Norwood, NJ: Ablex Publishing.

Traub, R. E., & MacRury, K. (In press). Multiple-choice vs. free-response in the testing of scholastic achievement. In K. Ingenkamp (Ed), Yearbook on educational measurement. Weinheim: Beltz Publishing Company.

Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. Psychometrika, 38, 1-10.

Ward, W. C., Frederiksen, N., & Carlson, S. B. (1980). Construct validity of free-response and machine-scorable forms of a test. Journal of Educational Measurement, 17, 11-29.

Table 1

Hypothesized Factor Model

	Factor	
Marker Variables	Multiple Choice	Free Response
Multiple Choice-A (10)	*	0
Multiple Choice-B (10)	*	0
Multiple Choice-C (10)	*	0
Multiple Choice-D (10)	*	0
Multiple Choice-E (10)	*	0
Free Response-A (1)	0	*
Free Response-B (1)	0	*
Free Response-C (1)	0	*
Free Response-D (1)	0	*
Free Response-E (1)	0	*

Note. The number of items per indicator is in parentheses.

Table 2

Means and Standard Deviations of APCS

Scores for Study Samples and the APCS Population

Score	Score Scale	Sample 1 Mean & SD (N=1000)	Sample 2 Mean & SD N=(1000)	Population Mean & SD (N=7372)
50-item Objective	0-50	26.3 (9.0)	26.0 (8.6)	26.2 (8.8)
Free-response #1	0-9	4.8 (3.6)	4.7 (3.5)	4.7 (3.5)
Free-response #2	0-9	6.0 (2.7)	5.8 (2.8)	6.0 (2.7)
Free-response #3	0-9	2.0 (2.8)	1.9 (2.8)	2.0 (2.9)
Free-response #4	0-9	2.1 (2.9)	2.0 (2.8)	2.0 (2.8)
Free-response #5	0-9	1.7 (2.5)	1.4 (2.3)	1.5 (2.4)

Note. The APCS 50-item objective score is calculated using number-right raw score.

Table 3

Factor Loadings for the Two-Factor Solution

Sample 1 (N=1000)		
	Factor	
Marker Variables	Multiple Choice	Free Response
Multiple Choice-A	.83	.00
Multiple Choice-B	.80	.00
Multiple Choice-C	.80	.00
Multiple Choice-D	.75	.00
Multiple Choice-E	.81	.00
Free Response-A	.00	.61
Free Response-B	.00	.70
Free Response-C	.00	.69
Free Response-D	.00	.61
Free Response-E	.00	.66

Sample 2 (N=1000)		
	Factor	
Marker Variables	Multiple Choice	Free Response
Multiple Choice-A	.80	.00
Multiple Choice-B	.80	.00
Multiple Choice-C	.82	.00
Multiple Choice-D	.71	.00
Multiple Choice-E	.77	.00
Free Response-A	.00	.60
Free Response-B	.00	.68
Free Response-C	.00	.67
Free Response-D	.00	.61
Free Response-E	.00	.65

Note. All loadings are significant at the .001 level (z-range for sample 1 = 20.30 to 31.37; z-range for sample 2 = 19.38 to 30.42).

Table 4

Comparison of Hypothesized and Alternative Factor Models

Sample and Factor Model	Fit Index			
	Chi-square/ df ratio	NNFI	AODASR	Akaike Information Criterion
Sample 1 (N=1000)				
Two-factor	3.68	.98	.02	57.17
One-factor	3.89	.98	.02	65.63
Null	117.47	---	---	5196.33
Sample 2 (N=1000)				
Two-factor	3.18	.98	.02	40.03
One-factor	4.25	.97	.02	78.60
Null	104.30	---	---	4603.66

Note. NNFI = non-normed fit index; AODASR = average off-diagonal absolute standardized residual.

Table 5

Hierarchical Chi-Square Tests of Competing Factor Models

Model Contrast	Chi-Square Model #1	Chi-Square Model #2	df Model #1	df Model #2	Chi-Square Diff	df Diff	p
Sample 1 (N=1000)							
2- vs. 1-factor	125.17	135.63	34	35	10.46	1	<.01
1-factor vs. Null	135.63	5286.33	35	45	5150.70	10	<.01
Sample 2 (N=1000)							
2- vs. 1-factor	108.03	148.60	34	35	40.57	1	<.01
1-factor vs. Null	148.60	4693.66	35	45	4545.06	10	<.01

Note. Model #1 is the more complex of the two models in a given contrast.

Table 6

Factor Loadings for the One-Factor Solution

Sample 1 (N=1000)	
Marker Variables	Loading
Multiple Choice-A	.82
Multiple Choice-B	.81
Multiple Choice-C	.80
Multiple Choice-D	.75
Multiple Choice-E	.80
Free Response-A	.60
Free Response-B	.70
Free Response-C	.67
Free Response-D	.60
Free Response-E	.65

Sample 2 (N=1000)	
Marker Variables	Loading
Multiple Choice-A	.80
Multiple Choice-B	.79
Multiple Choice-C	.81
Multiple Choice-D	.71
Multiple Choice-E	.77
Free Response-A	.57
Free Response-B	.66
Free Response-C	.64
Free Response-D	.58
Free Response-E	.62

Note. All loadings are significant at $p < .001$ level (z-range for sample 1 = 20.10 to 31.24; z-range for sample 2 = 19.06 to 30.16).

Figure Captions

1. Multiple-choice items. Copyright (c) 1988 by Educational Testing Service.

2. Free-response items. Copyright (c) 1988 by Educational Testing Service.

1. A program is being designed to enable users who are not computer experts to solve problems using a large file of geographic data: for example, to list the three longest rivers in Africa or to list the provinces of France. Of the following, which would be the most reasonable design for the user interface for such a program?

(A) Printing out a copy of the file
(B) Displaying the first screenful of the file, and then displaying the next screenful each time the user types a space
*(C) Displaying a menu of general topics on the screen and having the user proceed to lower-level menus by typing a single character
(D) Prompting the user to type an integer code for the data wanted
(E) Offering the user an optional tutorial that is designed to increase the user's expertise with computers in general

5. If evaluating *BBB* has no side effects, under what condition(s) can the program segment

while *BBB* **do**

Block1

be rewritten as

repeat

Block1

until not *BBB*

without changing the effect of the code?

(A) Under no conditions
(B) If executing *Block1* does not affect the value of *BBB*
*(C) If the value of *BBB* is true just before the segment is executed
(D) If the value of *BBB* is false just before the segment is executed
(E) Under all conditions

9. A list of integers can be stored sequentially in an array. The list can be maintained in sorted order. Maintaining the list in sorted order in an array leads to inefficient execution for which of the following operations?

I. Inserting and deleting elements
II. Printing the list
III. Computing the average of the elements

*(A) I only
(B) II only
(C) III only
(D) I and III only
(E) I, II, and III

13. The following program segment is intended to sum $A[1]$ through $A[N]$.

```
Sum := 0 ;
i := 0 ;
while i <> N do
  begin
    Sum := Sum + A[i] ;
    i := i + 1
  end
```

In order for this segment to perform as intended, which of the following modifications, if any, should be made?

(A) No modification is necessary.
(B) The segment *Sum* := 0 ; *i* := 0 ; should be changed to *Sum* := *A*[1] ; *i* := 1 ;
(C) The segment while *i* <> *N* do should be changed to while *i* <= *N* do
(D) The segment *Sum* := *Sum* + *A*[*i*] ; should be changed to *Sum* := *Sum* + *A*[*i* − 1] ;
* (E) The segment *i* := *i* + 1 should be interchanged with *Sum* := *Sum* + *A*[*i*]

2. Elapsed time is conventionally characterized in terms of three quantities: hours, minutes, and seconds. A type, *ElapsedTimeType*, could be implemented either as a single integer (elapsed seconds) or as three integers (elapsed hours, minutes, seconds) stored as a record with three integer fields.

   (a) Suppose that input, output, and arithmetic operations for variables of type *ElapsedTimeType* are to be implemented. Choose one of the two implementations of *ElapsedTimeType* and list the advantage(s) and disadvantage(s) of that choice.

   (b) Write type and variable declarations for the implementation chosen in part (a).

   (c) For the implementation chosen in part (a), write a procedure *PrintTime* that has one parameter of type *ElapsedTimeType* and that writes the value of its parameter in conventional form

   hh mm ss

   where *hh* is the number of hours of elapsed time, *mm* is the number of minutes (in addition to *hh* hours) of elapsed time, and *ss* is the number of seconds (in addition to *hh* hours and *mm* minutes) of elapsed time.

   (d) For the implementation chosen in part (a), write a procedure *TimeSum* that sets its third parameter to the sum of its first two parameters. All three of the parameters are to be of type *ElapsedTimeType*.

4. Write a procedure that reverses the order of the elements of a linked list pointed to by the parameter of the procedure. The list is implemented using the following declarations.

```
type
  PtrNode = ^NodeType ;
  NodeType = record
               Data : integer ;
               Next : PtrNode
             end ;
```

The procedure you are to write is to have the following header.

```
procedure Reverse(var Head : PtrNode) ;
```

Appendix A:

Example Scoring Rubrics

Source: The 1988 Advanced Placement Examinations in Computer Science and their grading. New York: College Entrance Examination Board, 1989.

Scoring Rubric for Free-Response Problem #2

Traditionally, each part of a question is always worth the same number of points, but this was a new kind of question, so we introduced a new kind of rubric. Depending on the implementation chosen, parts (c) and (d) were worth a different number of points.

Both implementations were graded as follows for parts (a) and (b):

- +2 properties of implementation chosen (part a)
  - +1 one valid advantage
  - +1 one valid disadvantage
- +1 perfect declaration of *ElapsedTimeType* (part b)

Contrary to the usual grading practice, if students listed more than one advantage/disadvantage in (a), we graded the best one, even if one or more of the others were actually incorrect. In this particular case, we decided that there was enough complexity in the problem to justify such leniency.

Parts (c) and (d) were graded as follows for the simple integer solution:

- +4 implementation of *PrintTime* (part c)
  - +1 procedure header (could be lost in usage)
  - +1 properly printing hours (somehow extracting them from simple integer)
  - +1 properly printing minutes (somehow extracting them from simple integer)
  - +1 properly printing seconds (somehow extracting them from simple integer)
- +4 implementation of *TimeSum* (part d)
  - +1 procedure header with parameters referenced in body (could be lost in usage)
  - +1 statement of the form *result* := *t1* + *t2*

Parts (c) and (d) for the record implementation were graded as follows:

- +2 implementation of *PrintTime* (part c)
  - +1 procedure header (could be lost in usage)
  - +1 properly prints hours, minutes, and seconds
- +4 implementation of *TimeSum* (part d)
  - +1 procedure header with parameters referenced in body (could be lost in usage)
  - +1 computes seconds, with overflow to minutes
  - +1 computes minutes, with overflow from seconds to hours
  - +1 computes hours, with overflow from minutes

Students were expected to create an appropriate header form the informal specification. In particular, all parameters were to be of type *ElapsedTimeType* and students were expected to appropriately distinguish **var** and value parameters (i.e., *PrintTime* with one **var**, *TimeSum* with its result **var** and the other two parameters value). Students who used value instead of **var** lost 1 point in usage. Students who used **var** instead of value lost only ½ point in usage, because although they used the wrong kind of parameter, the code still works.

Some special rules for grading (d) were introduced for the record implementation so as not to penalize students twice for the same mistake. For example, a student who tested for (*seconds* > *60*) and (*minutes* > *60*) rather than (*seconds* >= *60*) and (*minutes* >= *60*) would lose only 1 point, not 2.

The most common credited advantages and disadvantages of the two implementations are summarized below. In most cases, an advantage of one implementation becomes a disadvantage of the other.

Reason	integer	record
clarity of code	advantage	advantage
efficient space usage	advantage	disadvantage
easy coding of arithmetic operators	advantage	disadvantage
easy coding of I/O operators	disadvantage	advantage
easy to get elapsed seconds	advantage	disadvantage
approach is intuitive	disadvantage	advantage
approach works better for large values	disadvantage	advantage
approach models real world	disadvantage	advantage

Scoring Rubric for Free-Response Problem #4

All iterative solutions were graded with the following rubric:

- +7 reversing links
  - +1 tests for empty list
  - +1 correctly handles list of length 1
  - +4 correctly handles list of length > 1
    - +2 correctly rearranges middle elements
    - +2 properly handles both endpoints
  - +1 *head* properly reassigned
- +2 efficiency: $O(n)$ time, $O(1)$ auxiliary space (+1 for $O(n)$ auxiliary storage that is *disposed* before procedure terminates)

Here are some examples of what efficiency points would be taken off for several situations. As always, no more than 2 can be taken off even if more than one of them applies.

Situation	Penalty
single pass through list	none
several passes through list, but independent of length	none
multiple traversals of list leading to (n^2) time	–2
duplicate structure using *new*, disposing inside loop	none
duplicate structure using *new*, disposing after loop	–1
secondary array	–2
secondary stack implemented as array	–2
duplicate structure using *new*, no disposing	–2

Recursive solutions were graded by Table Leaders. Solutions that wrote the contents of the list in reverse order, but never rearranged the data stored in the list, received no credit.

Appendix B:

Sample Correlation Matrices

Sample 1 (N=1000)										
	1	2	3	4	5	6	7	8	9	10
1. Multiple Choice-A	2.33									
2. Multiple Choice-B	.66	2.22								
3. Multiple Choice-C	.66	.67	2.08							
4. Multiple Choice-D	.61	.60	.61	1.90						
5. Multiple Choice-E	.70	.64	.63	.59	2.20					
6. Free Response-A	.49	.51	.48	.45	.44	3.57				
7. Free Response-B	.57	.59	.55	.50	.56	.51	2.74			
8. Free Response-C	.55	.51	.52	.50	.56	.39	.45	2.79		
9. Free Response-D	.46	.47	.47	.48	.47	.35	.38	.44	2.92	
10. Free Response-E	.52	.49	.51	.51	.52	.38	.40	.51	.48	2.50

Sample 2 (N=1000)										
	1	2	3	4	5	6	7	8	9	10
1. Multiple Choice-A	2.22									
2. Multiple Choice-B	.64	2.13								
3. Multiple Choice-C	.66	.65	2.07							
4. Multiple Choice-D	.54	.57	.60	1.91						
5. Multiple Choice-E	.63	.62	.61	.55	2.05					
6. Free Response-A	.45	.44	.47	.38	.42	3.53				
7. Free Response-B	.50	.54	.51	.49	.51	.47	2.76			
8. Free Response-C	.52	.48	.51	.45	.48	.36	.40	2.81		
9. Free Response-D	.45	.42	.44	.40	.43	.35	.38	.43	2.82	
10. Free Response-E	.47	.46	.49	.43	.46	.34	.40	.48	.50	2.27

Note. Standard deviations are presented on the diagonal.