RR-89-33

RESEARCH REPORT

THE RELATIONSHIP OF CONSTRAINED FREE-RESPONSE TO MULTIPLE-CHOICE AND OPEN-ENDED ITEMS

Randy Elliot Bennett
Donald A. Rock
Henry I. Braun
Douglas Frye
James C. Spohrer
Elliot Soloway

Educational Testing Service
Princeton, New Jersey
June 1989

The Relationship of Constrained Free-Response to Multiple-Choice and Open-Ended Items

Randy Elliot Bennett
Donald A. Rock
Henry I. Braun
Educational Testing Service

Douglas Frye
James C. Spohrer
Yale University
and
Elliot Soloway
University of Michigan

Acknowledgements

Appreciation is expressed to Minh Wei Wang and Bruce Kaplan for their assistance in data analysis. Hazel Klein and Terri Stirling were instrumental in organizing and managing the data collection effort. Thanks are due to Carl Haag of the AP program and to C. Victor Bunderson for their encouragement and support. Finally, we are indebted to the students and teachers of the Advanced Placement Program without whom this study would not have been possible.

Abstract

This study examined the relationship of a machine-scorable, constrained free-response computer science item that required the student to debug a faulty program to two other types of items: (1) multiple-choice and (2) free response requiring production of a computer program. Confirmatory factor analysis was used to test the fit of a three-factor model to these data and to compare the fit of this model to three alternatives. These models were fit using two random-half samples, one given a faulty program containing one bug and the other a program with three bugs. A single-factor model best fit the data for the sample taking the 1-bug constrained free response and a two-factor model fit the data for the second sample. In addition, the factor intercorrelations showed this item type to be significantly related to both the free-response items and the multiple-choice measures.

The Relationship of Constrained Free-Response to Multiple-Choice and Open-Ended Items

Over the better part of a century, the multiple-choice item has been the mainstay of standardized testing in the United States. The use of this format is justified by its objectivity and efficiency, and more recently by the development of a strong statistical foundation for its analysis (e.g., Lord, 1980).

Multiple-choice items have, however, been criticized because they often do not directly resemble criterion behaviors, are of limited utility for instructional diagnosis, and might not be capable of measuring certain cognitive processes or skills. To address these limitations, a heavier reliance on constructed response (e.g., essays, performance tasks) is often suggested. Constructed response items can present tasks similar to those encountered in education and work settings, offer a window onto problem solving processes (Birenbaum & Tatsuoka, 1987), and may measure somewhat different skills than multiple-choice formats (Ward, Frederiksen, & Carlson, 1980).

Whereas constructed response formats offer attractive potential advantages, their main liabilities for major testing programs have been the subjectivity and high cost associated with scoring. For example, the College Board's Advanced Placement Program annually invests substantial resources to gather and house several hundred teachers who score tens of thousands of constructed responses. Although significant efforts are made to enhance objectivity (e.g., teachers are trained to score each question and two levels of re-reading occur for samples of papers), variation across readers is at times considerable (Braun, 1988). If a machine-scorable constructed-response item type could be developed, problems associated with scoring cost and reliability might be substantially reduced.

One example of progress toward developing such an item type is in computer science (Braun, Bennett, Soloway, & Frye, in press). This item type presents the examinee with a specification describing a task to be performed by a computer program and a completed program that does not correctly perform that task. It is the examinee's assignment to correct the program by deleting and/or inserting the required code. The student's corrected program is then given to an expert system for scoring. In a recent study (Braun et al., in press), this experimental system was able to produce a score for 83% of the papers it encountered and agreed with a human rater at levels similar to those at which raters agree among themselves (product-moment correlations in the eighties).

The purpose of this study was to assess the relationship of this constrained, free-response item type to multiple-choice and to free-response items contained on the College Board's Advanced Placement Computer Science Examination. This relationship dictates the potential of this new item-type as a replacement for more open-ended formats and as a supplement to multiple-choice items.

Method

Subjects

Subjects were drawn from a prior study of the item type conducted with a sample of high school seniors taking the 1988 APCS examination (Braun et al., in press). Subject selection procedures involved (1) soliciting participation from all APCS teachers with class enrollments of 15 or more or who had participated in grading the 1987 APCS examination, (2) receiving indications of interest from teachers at 70 of 112 solicited schools, (3) mailing constrained free-response items to these teachers, (4) receiving responses from 916 students in 59 schools, and (5) locating in ETS files 1988 APCS scores for 737 of these students for whom responses were judged to be complete. Of these 737 completed records, the constrained, free-response item type was able to be machine-scored for 614 students. For purposes of this study, this sample was split into (approximately) random halves, differentiated by having taken variants of the faulty solution problem that contained 1 vs. 3 bugs.

Instruments

Constrained free-response item. The constrained free-response item was a more structured adaptation of an open-ended problem from the 1985 APCS examination. The open-ended version required the student to write a program that rotates the elements of an array. Eight constrained variants of this problem were developed as a means of increasing the breadth of the content domain studied. Each variant contained a program specification and a faulty solution to that specification. In six of the variants, the solution contained a single bug; in two variants, three bugs each were embedded with each bug chosen to avoid interactions with other bugs.

Bugs were chosen to reflect three categories that have been found to capture most of the nonsyntactic errors produced by novices when writing programs (Spohrer, 1989). These categories were arrangement, completeness, and detail. An arrangement bug occurred when all of the parts of a program were present but not put together properly. A completeness bug existed when one component was missing. When a single part of a component was at fault (e.g., a variable, operator) and could be repaired by changing one word or operator, the bug fell into the last category.

Two bugs were selected from each category, for a total of six different bugs (one for each single-bug variant). Each of the triple bug variants contained one bug from each category. One variant, along with directions, is presented in the appendix.

Students' responses to these items were presented to the expert system, MicroPROUST (Johnson & Soloway, 1985), in the context of the full program that the student was to correct. MicroPROUST scores solutions by (1) breaking a problem down into a set of component goals, (2) comparing sections of the student's program to correct ways of achieving those goals, and where it can't find a match (3) comparing those sections with common faulty implementations of the goals. On the basis of the faults detected, diagnostic comments are produced and numeric scores are assigned on a 0 to 2 scale. Rater reliability was computed by correlating expert system scores with those of a human grader. For the 1-bug variants, the correlation was .88 (n = 40) and for the 3-bug variants .82 (n = 44) (Braun et al., in press).

The Advanced Placement Computer Science Examination. Two Advanced Placement Computer Science Examinations are offered: an "A" exam intended to assess mastery of topics covered in the first semester of a college-level introductory course in computer science, and an "AB" exam covering the full year's material. Computer Science "A" is included in its entirety in the "AB" examination so that students completing the full year's course also take the "A" examination. Because more students take the "A" exam, it is used in this study. Computer Science "A" emphasizes programming methodology and procedural abstraction, but also includes some material on the study of algorithms, data structures, and data abstraction. This exam is comprised of 35 multiple-choice and 3 free-response items (see appendix for examples). The free-response items, which are scored by human graders, 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.

Procedure

Each student was asked to respond to one of the eight problem variants and to one of eight variants for a second problem. Responses to the second problem were not included in this study because they were scored by a second expert system for which rater reliability was found to be suspect. Variants were randomly assigned to students such that equal numbers of 1- and 3-bug versions were administered. Teachers were instructed to administer the problems in a single class period during the month prior to the APCS examination. Problems were given in paper-and-pencil format with allowable modifications limited to insertions and deletions. Upon receipt, solutions were converted to machine-readable format in the course of which obvious and minor syntax errors were corrected.

A three-factor model composed of multiple-choice, free-response, and faulty-solution factors was posed to test the relationship of the new item type to the two others. The factors composing the hypothesized model were marked by the three item types. For the first factor, these item types were parcels of APCS multiple-choice items balanced on difficulty. Three multiple-choice parcels were constructed from every third item in each of four test specification content areas (programming methodology, features of languages, algorithms, and computer systems)--and a single item from each of two additional areas (data structures and applications). Items were then shifted among parcels (but within content categories) so that the mean difficulty values for each parcel were similar. Parcels were scored on a 12- or 13-point number-right scale based on the number of items in the parcel. The second factor was indicated by each of three APCS free-response problems, with each free-response scored on a ten-point scale. Finally, the third factor was marked by a single indicator, the response to the "Rotate" problem. This problem was scored on a five-point scale for the sample taking the 1-bug variants and a six-point scale for the group taking the 3-bug versions.

Table 1 depicts 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 particular factor. To estimate the factor pattern from the data, the sample polychoric correlation matrix was computed using the PRELIS program (Joreskog & Sorbom, 1986). The weighted least squares factor estimation procedure from LISREL 7 (Joreskog & Sorbom, 1988) was then used to estimate the unknown factor loadings (i.e., the asterisks) subject to the pattern of zero constraints and allowing the factors to be intercorrelated.

Insert Table about 1 here

The factor pattern was estimated from the polychoric correlation matrix using the weighted least squares procedure because the scales for the marker variables were more or less restricted and the resulting distributions non-normal. The weighted least squares procedure provides for asymptotic standard errors and overall goodness-of-fit tests that do not assume normality.

To estimate accurately the relationship between factors, a reliability estimate for each factor must be available. For factors with multiple markers, this estimate is generated from within the factor model. Because there was only one indicator of the constrained free-response factor, the reliability of this factor could not be estimated in this way. Hence, an external estimate was needed.

To approximate the reliability of the faulty-solution item, the average reliability of the free-response items was used. This reliability estimate can be argued to be a lower bound for the faulty solution because the free-response estimate includes two sources of variation: topic (each problem poses a different task) and rater (each solution is graded by a different individual). The faulty solution is computer scored; thus, there is no rater variance, leaving topic as the only source of variation. To compute the reliability estimate, the factor loadings for the model were estimated, the loading for each free response in the weighted least squares solution was squared, and these squared loadings were averaged. The resulting reliabilities were .56 for sample 1 and .62 for sample 2. Finally, the solutions were re-run using these estimates for the reliability of the faulty solutions.

The fit of the three-factor model was assessed by examining its factor intercorrelations and goodness-of-fit indicators, and by comparing the model's fit to several reasonable alternatives. The alternative models were (1) a null model in which no common factors were presumed to underlie the data (i.e., each of the seven markers was allowed to load only on its own factor), (2) a general model in which all variables loaded on a single factor, (3) a two-factor solution composed of APCS test and constrained free-response factors intended to assess whether the constrained responses were measuring attributes different from the test, and (4) a three-factor model restricting each item type to load on a separate 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.

Evaluating model fit was complicated by the fact that, in confirmatory factor analysis, universally accepted measures of fit do not exist (Marsh & Hocevar, 1985; Sobel & Bohrnstedt, 1985). Consequently, several goodness-of-fit indicators were used, particularly in comparing the three-factor model to the alternatives. These indicators were:

Tucker-Lewis index. The Tucker-Lewis (T-L) index (Tucker & Lewis, 1973) represents the ratio of the variance associated with the model to the total variance, and may be interpreted as indicating how well a factor model with a given number of common factors represents the covariances among the markers. A low coefficient indicates that the relations among the markers are more complex than can be represented by that number of common factors.

Root means square residual. The root mean square residual (RMSR) is the average correlation among the markers that is left over after the hypothesized model has been fitted (Joreskog & Sorbom, 1988). The lower the RMSR, the better the fit.

Chi-square/degrees of freedom ratio. The chi-square/degrees of freedom ratio is based upon the overall chi-square goodness-of-fit test associated with each factor model. Ratios up to 5.0 indicate a reasonable fit (Marsh & Hocevar, 1985).

Goodness-of-Fit index. Ranging from 0 to 1.00, the Goodness-of-Fit index (GFI) is a measure of the relative amount of variance and covariance jointly accounted for by the factor model (Joreskog & Sorbom, 1988). The higher the magnitude of this index, the better the model fit.

Akaike information criterion. The Akaike information criterion (AIC) is an index of parsimony in which the best fitting model is defined as having a small chi-square with few unknowns (Loehlin, 1987). As scaled here, the AIC is always negative, with the best fitting model having the index closest to zero.

Standardized residuals. Standardized residuals can be used to judge fit and to locate the specific causes of a lack of fit. In general, residuals larger than 2.0 in magnitude suggest a problem with the model (Joreskog & Sorbom, 1988).

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.) Also presented are the summary statistics for performance on the faulty solution items for the two study samples. For each APCS score, a two-tailed *z*-test was used to contrast each sample mean with the population mean, which was treated as a population parameter. While the sample means proved to be significantly higher than the test population mean for most contrasts, the magnitude of these differences was marginal ranging from .14 to .26 standard deviations. These marginal differences suggest that the samples were not dramatically different in computer science knowledge from the population taking the examination.

Insert Table 2 about here

Table 3 presents the loadings for each variable as estimated from the three-factor model. In both samples, all loadings are highly significant ($\underline{p} < .001.$, $\underline{t}$ range 14.01 to 39.95). Loadings for the multiple-choice factor are generally slightly higher than those for the free-response factor, probably due to the fact that the multiple-choice indicators were constructed so as to be parallel in content and difficulty. Hence, these indicators share a great deal of variance. In contrast, each free-response indicator 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

The absolute fit of the three-factor model can be evaluated through inspection of several indices. The goodness-of-fit indices and standardized residuals suggest the extent to which the model is complex enough to account for the data. For samples 1 and 2, the T-L index was 1.00 and .99, respectively, indicating that the three-factor model accounts for virtually all of the variance among the markers. The RMSRs--which indicate the average correlation among the markers left over after the three-factor model is fitted--present a similar picture: .02 for both samples. Third, inspection of the standardized residuals reveals that none were larger in magnitude than 2.0 in sample 1 and only one of 28 was larger than 2.0 in sample 2, a finding expected on the basis of chance alone.

Factor intercorrelations suggest whether a simpler model might account for the data. Table 4 gives the factor intercorrelations for the three-factor model. For sample 1 (which took the 1-bug variants), the disattentuated correlations are so high as to question the need for a three-factor model. For sample 2 (which took the 3-bug variants), the correlations between the constrained free-response factor and the other factors are lower, though that between free-response and multiple-choice is high enough to suggest the need for a simpler model.

Insert Table 4 about here

The fit of the three-factor model in relation to several more parsimonious alternatives is presented in Table 5. For sample 1, negligible losses in fit occur for most indexes in moving from the three- to the single-factor solutions. The changes are, however, substantial once the null model is reached. For example, the RMSR remains the same from the three-factor to the single-factor models, but increases by .49 from the single-factor to the null models. In contrast to the other indices, the Akaike information criterion--a measure of parsimony--shows marginal improvements in fit through the single-factor solution.

For sample 2, the pattern is similar. The largest losses are associated with the move from the single-factor to the null models, and most indices show only trivial changes from the three- to the one-factor solutions. A hint of a slightly better fit for the two- over the one-factor model, however, is given by the Akaike information criterion, which is at its lowest for the two-factor solution.

Insert Table 5 about here

The relative fit of the models can also be assessed by examining the distributions of the standardized residuals (see Table 6). For sample 1, the residuals change marginally from the 3-factor to the single-factor solutions, but become dramatically larger when the null model is reached. For sample 2, a comparable pattern is displayed.

Insert Table 6 about here

This suggestion of a reasonable fit for the single factor model in sample 1 and possibly the two-factor model in sample 2 can be further evaluated by inspecting the intercorrelations from the two-factor model. For sample 1, the disattentuated correlation is .93 ($\underline{p}$ < .001, $\underline{t}$ = 12.09), too high to support a two-factor solution; for sample 2, it is .71 ($\underline{p}$ < .001, $\underline{t}$ = 11.92), a value more consistent with a two-factor model.

Table 7 shows the loadings for the two-factor solution. Again, all loadings are highly significant ($\underline{p}$ < .001; $\underline{t}$ range = 14.01 to 40.27). As for the three-factor solution, the loadings for the multiple-choice markers are slightly higher than those for the free-responses. The probable explanation is similar: being parallel, the multiple-choice markers share more variance and, as a result, play a bigger role in defining the common factor than do the free-response indicators.

Insert Table 7 about here

Discussion

This study examined the relationship of one form of a constrained, free-response item type--faulty solutions--to multiple-choice and to free-response items contained on the College Board's Advanced Placement Computer Science Examination. Results suggested that the three item types formed a single factor in one sample but that a two-factor model with the faulty solutions defining a separate factor might better account for the data in the second sample. Further, examination of the factor intercorrelations indicated that the faulty solutions were significantly related to both the free-response items and the multiple-choice measures.

What might account for the differences in fit between the two samples? One potential explanation is that the timing guidelines under which the items were administered allotted less time per bug to those taking the 3-bug problems. This differential might have created a power vs. speed situation in which the major source of individual differences among students taking the 1-bug variants was programming skill whereas for those taking the 3-bug variants, speed of processing might also have been called into play.

In addition to the variation in factor structure across samples, the finding that the faulty solutions were significantly related to the free-response items is of interest. This result, which occurred in both samples, suggests that the premise for the constrained free-response format is plausible: to combine in a single item type the surface characteristics and cognitive demands of free response with the machine-scorable efficiency of multiple choice. That faulty solutions might be reliably machine-scored is supported by a companion investigation which found that most student responses could be analyzed and that scores generally were similar to those awarded by a human grader (Braun, Bennett, Frye, & Soloway, in press).

Whereas faulty solutions were significantly related to free-response items, faulty solutions were also significantly correlated with multiple-choice questions. This affinity for both item types is seemingly owed to the exceptionally high relationship observed between multiple-choice and free-response items. This latter result would appear to be a stable one since correlational analyses of student performance on other forms of the APCS examination with different samples have produced the same finding (Mazzeo & Flesher, 1985; Mazzeo & Bleistein, 1986; Bleistein, Maneckshana, & McLean, 1988). Similar relationships between multiple-choice and constructed-response formats have been reported in other content areas, specifically mathematical reasoning (Traub & Fisher, 1977) and verbal reasoning (Ward, 1982), though such a result is not universal (e.g., Ackerman & Smith, 1988; Ward, Frederiksen, & Carlson, 1980).

One likely reason for the present finding is that in some situations free-response and multiple-choice items may measure the same processes. Traub and Fisher (1977) make such an argument for mathematical reasoning in which they suggest that the examinee must construct a solution regardless of the item format, though in the multiple-choice case the resulting answer is used as a basis for choosing among the response options. (Note, however, that locating one's constructed answer among the options is no guarantee that the answer is correct.)

In the APCS context, this argument would appear to have some merit. For example, a cursory analysis of multiple-choice item content suggests that many of these items cannot be correctly answered with any consistency and efficiency by strategies other than construction, in which case the processes used would arguably be identical or highly similar to those employed in writing a program or design. These items call for such things as choosing the correct data structure, counting loop executions, and finding bugs.

This explanation of shared processes may not, however, be entirely satisfactory. One reason is that some items explicitly require the examinee to recognize in a set of response options the one that best satisfies some condition, where the set of potential correct options is too large to justify generating a response before consulting the listed options. That is, the <u>only</u> efficient strategy is to read each response option and determine if it does or does not satisfy the condition. This recognition process is arguably different from the recall processes exhibited in constructing a program (or in answering some of the other multiple-choice items). A second reason why it may not be safe to assume that the APCS multiple-choice and free-response items call entirely upon the same processes is that some multiple-choice items ask for simple factual recognition and, sometimes, not even in the programming domain (e.g., one that asks about the defining characteristics of a compiler). Finally, there are probably processes not well-represented in the multiple-choice section that are called for in writing a program (e.g., planning, synthesis).

If these contentions are true, how can we account for the virtually perfect correlation between the multiple-choice and free-response factors? As noted, part of the covariation is probably due to shared processes. Much of the rest is plausibly owed to high relations among processes or to processes and knowledges that are developed together. For example, it is possible that the processes invoked in responding to multiple-choice and free-response items are correlated by virtue of being subcomponents of a single, more general ability (Sternberg, 1980). Or, it is plausible that some knowledges are developed because they are taught along with programming skill or develop incidentally as a result of it.

Further research might help resolve many of these conjectures. In particular, cognitive analyses of the tasks posed by the APCS multiple-choice and free response items, and by the faulty solutions, might better elucidate the degree to which these item types measure different processes. Such analyses might also identify how single and multiple-bug faulty-solutions tasks differ. Second, studies of the functioning of the faulty-solutions item type in other domains (e.g., algebra word problems) should help identify whether and how this format might be used in assessing skills other than programming. Finally, development of a prototype intelligent assessment system might be explored. In such a system, multiple-choice items would be presented first. The information from these items would then be used to determine whether to present constructed-response items (i.e., faulty solutions and/or free-response) to a given student and also to leverage the expert system's interpretation of the student's answers. This combination of student screening and leverage might allow the level of successful analyses of constructed responses to approach 100%.

Several limitations of the present study should be noted. First, the use of only a single instance of the constrained free-response item type within each sample is a weakness. Though multiple variants were employed, using only a single problem limits greatly the generalizability of results to faulty solutions as a class of constrained free-response as well as to other classes of constrained free-response (e.g., completion items). Further, using a single exemplar prevented a reliability estimate for the item type from being generated by the factor model, forcing the estimate to be approximated with the reliability of the free response items. While this is argued to be a reasonable approximation, it is upon this approximation that the intercorrelations between the constrained free-response and other factors are based. If, for example, this approximation is too low, the corrected intercorrelations may be too high. Future studies should include multiple instances to increase the likelihood of yielding accurate estimates and the generalizability of results.

A second limitation is that the effects of item format could not be strictly tested because content was not held constant across formats. That is, different problems were presented in the three formats. (As noted above, in some cases, multiple-choice problems did not even deal directly with programming skill.) However, even with these content differences the formats were highly intercorrelated (with the exception of the 3-bug faulty solution).

Third, all measures were not given at the same point in time. Whereas the APCS multiple-choice and free-response problems were administered on the same day, the faulty solutions were given up to a month before, though exactly when within this period differed among the participating schools. It is possible that some relevant learning might have occurred between the two administrations. However, as both the 1- and 3-bug variants were administered within each school, additional learning (or other variables related to time between administrations) does not seem a plausible explanation for the observed differences in factor structure.

Finally, even though the faulty-solutions and free-response tasks involved construction, they are still somewhat removed from classroom debugging and programming behaviors. In the classroom, both behaviors are performed interactively, not in the paper-and-pencil mode employed in this study. Whether interactive environments that allowed examinees to execute the programs they were writing or debugging would still produce factor structures like those found here is an unresolved question.

What are the implications of this study for the APCS examination? If our results can be replicated with faulty solutions covering a wider range of programming skill, an argument might be made for eventually including the 1-bug variant in the current test. Substituting several faulty solutions for a free-response question would apparently not change the essential construct measured by the test and might possibly reduce scoring costs over the long term. This cost reduction is by no means assured: substantial effort is required to develop the knowledge base needed to score responses to each faulty solution and it is not yet clear how much a problem can be changed before major modifications in the knowledge base need to be made. With respect to the 3-bug faulty solution, a better understanding of the role of time limits and of any potential differences in cognitive requirements is required before use of this version can be seriously considered.

Finally, even though multiple-choice and free-response appear to measure the same essential APCS construct, there are good reasons to maintain--and perhaps increase--the role of constructed-response items. The most compelling reason is that the ability to successfully complete free-response items --that is, to program--is central to the APCS curriculum. Including free-response items emphasizes to teachers and students the need to focus on developing this skill. Second, the multiple-choice format is viewed by many testing critics as measuring and encouraging the development of irrelevant skills. The inclusion of constructed-response items should help respond to these concerns, thereby increasing the credibility of our measures and their results.

References

Ackerman, T. A., & Smith, P. L. (1988). A comparison of the information provided by essay, multiple-choice, and free-response writing tests. Applied Psychological Measurement, 12, 117-128.

Bennett, R. E., Gong, B., Kershaw, R. C., Rock, D. A., Soloway, E., & Macalalad, A. (In press). Assessment of an expert system's ability to automatically grade and diagnose students' constructed-responses to computer science problems. In R. O. Freedle (Ed), Artificial intelligence and the future of testing, Hillsdale, NJ: Lawrence Erlbaum Associates.

Bleistein, C., Maneckshana, B., & McLean, D. (1988). Test analysis: College Board Advanced Placement Examination Computer Science 3JBP (SR-88-63). Princeton, NJ: Educational Testing Service.

Braun, H. I. (1988). Understanding scoring reliability: experiments in calibrating essay readers. Journal of Educational Statistics, 13, 1-18.

Braun, H. I., Bennett, R. E., Frye, D., & Soloway, E. (In press). Developing and evaluating a machine-scorable, constrained constructed-response item. Princeton, NJ: Educational Testing Service.

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.

Johnson, W. L., & Soloway, E. (1985). PROUST: An automatic debugger for Pascal programs. Byte, 10(4), 179-190.

Joreskog, K., & Sorbom, D. (1986). PRELIS: A program for multivariate data screening and data summarization. Mooresville, IN: Scientific Software, Inc.

Joreskog, K., & Sorbom, D. (1988). LISREL 7: A guide to the program and applications. Chicago, IL: SPSS Inc.

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

Lord, F. M. Applications of item response theory to practical testing problems. Hillsdale, NJ: Erlbaum.

McNemar, Q. (1962). Psychological statistics. New York: Wiley.

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.

Mazzeo, J., & Bleistein, C. (1986). Test analysis: College Board Advanced Placement Examination Computer Science 3IBP (SR-86-105). Princeton, NJ: Educational Testing Service.

Mazzeo, J., & Flesher, R. (1985). Test analysis: College Board Advanced Placement Examination Computer Science 3HBP (SR-85-180). Princeton, NJ: Educational Testing Service.

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.

Sternberg, R. J. (1980). Factor theories of intelligence are all right almost. _Educational Researcher_, _9_, 6-18.

Traub, R. E., & Fisher, C. W. (1977). On the equivalence of constructed-response and multiple-choice tests. _Applied Psychological Measurement_, 1, 355-369.

Tucker, L. R., & Lewis, C. (1973). A reliability coefficient for maximum likelihood factor analysis. _Psychometrika_, _38_, 1-10.

Ward, W. C. (1982). A comparison of free-response and multiple-choice forms of verbal aptitude tests. _Applied Psychological Measurement_, _6_, 1-11.

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	Constrained Free-Response
Multiple Choice-A (12)	*	0	0
Multiple Choice-B (12)	*	0	0
Multiple Choice-C (11)	*	0	0
Free Response-A (1)	0	*	0
Free Response-B (1)	0	*	0
Free Response-C (1)	0	*	0
Constrained Free-Response (1)	0	0	*

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

Table 2

Means and Standard Deviations of APCS and Faulty-Solution Scores for Study Samples and the APCS Population

Score	Score Scale	Sample 1 Mean & SD (N=314)	Sample 2 Mean & SD N=(300)	Population Mean & SD (N=10,719)
APCS				
35-item Objective	0-35	17.3**	17.8***	16.1
		(6.4)	(6.6)	(6.5)
Free-response #1	0-9	4.6*	4.8***	4.1
		(3.7)	(3.7)	(3.5)
Free-response #2	0-9	5.8**	5.9***	5.3
		(2.8)	(2.8)	(2.9)
Free-response #3	0-9	1.8	1.9	1.6
		(2.7)	(2.8)	(2.7)
Rotate				
1-bug variants	0-2	1.23	----	----
		(.94)	----	----
3-bug variants	0-2	----	.89	----
		----	(.69)	----

Note. All APCS scores are calculated using number-right raw score.

*_p_ < .05, two-tailed test of student sample mean with test population mean.
**_p_ < .01, two-tailed test of student sample mean with test population mean.
***_p_ < .001, two-tailed test of student sample mean with test population mean.

Table 3

Factor Loadings for the Three-Factor Model

Sample 1 (N=314)

	Factor		
Marker Variables	Multiple Choice	Free Response	Constrained Free-Response
Multiple Choice-A	.84	.00	.00
Multiple Choice-B	.81	.00	.00
Multiple Choice-C	.81	.00	.00
Free Response-A	.00	.69	.00
Free Response-B	.00	.77	.00
Free Response-C	.00	.77	.00
Constrained Free-Response	.00	.00	.75

Sample 2 (N=300)

	Factor		
Marker Variables	Multiple Choice	Free Response	Constrained Free-Response
Multiple Choice-A	.84	.00	.00
Multiple Choice-B	.83	.00	.00
Multiple Choice-C	.86	.00	.00
Free Response-A	.00	.75	.00
Free Response-B	.00	.77	.00
Free Response-C	.00	.82	.00
Constrained Free-Response	.00	.00	.79

Note. All loadings are significant at the .001 level (*t* range for sample 1 = 14.01 to 35.50; *t* range for sample 2 = 15.16 to 39.95). Sample 1 completed the 1-bug variants. Sample 2 completed the 3-bug variants.

Table 4

Factor Intercorrelations:

Three-Factor Solution

Sample 1 (N=314)	Multiple Choice	Free Response	Constrained Free-Response
Multiple Choice		.97	.89
Free Response			.98
Constrained Free-Response			

Sample 2 (N=300)	Multiple Choice	Free Response	Constrained Free-Response
Multiple Choice		.98	.68
Free Response			.74
Constrained Free-Response			

Note. All correlations are significant at *p* < .001 level (*t* range for sample 1 = 10.45 to 29.48; *t* range for sample 2 = 10.14 to 35.14). Sample 1 completed the 1-bug variants. Sample 2 completed the 3-bug variants.

Table 5

Comparison of Hypothesized and Alternative Factor Models

	Fit Index				
Sample and Factor Model	Chi-square/ df ratio	T-L Index	RMSR	GFI	Akaike Information Criterion
Sample 1 (N=314)					
Three-factor	.32	1.00	.02	1.00	-17.92
Two-factor	.48	.99	.02	1.00	-17.39
One-factor	.50	.99	.02	1.00	-16.72
Null	72.47	---	.51	.42	-767.96
Sample 2 (N=300)					
Three-factor	.48	.99	.02	1.00	-18.89
Two-factor	.51	.99	.02	1.00	-17.54
One-factor	1.30	.98	.03	.99	-22.72
Null	80.59	---	.52	.38	-853.14

Note. Sample 1 completed the 1-bug variants. Sample 2 completed the 3-bug variants.

Table 6

Frequency Distributions of Standardized Residuals

for Hypothesized and Alternative Factor Models

	Sample 1 (N=314)			
Standardized	Model			
Residual	Null	1-factor	2-factor	3-factor
>3	21			
>2 to 3				
>1 to 2		1	1	
-1 to 1	7	23	23	27
<-1 to -2		4	4	1
<-2 to -3				
<-3				

	Sample 2 (N=300)			
Standardized	Model			
Residual	Null	1-factor	2-factor	3-factor
>3	21			
>2 to 3				
>1 to 2			1	
-1 to 1	7	22	24	24
<-1 to -2		5	3	3
<-2 to -3		1		1
<-3				

Note. Sample 1 completed the 1-bug variants. Sample 2 completed the 3-bug variants.

Table 7

Factor Loadings for the Two-Factor Model

Sample 1 (N=314)		
	Factor	
Marker Variables	Multiple Choice	Constructed Response
Multiple Choice-A	.84	.00
Multiple Choice-B	.81	.00
Multiple Choice-C	.81	.00
Free Response-A	.68	.00
Free Response-B	.76	.00
Free Response-C	.76	.00
Constrained Free-Response	.00	.75

Sample 2 (N=300)		
	Factor	
Marker Variables	Multiple Choice	Constructed Response
Multiple Choice-A	.84	.00
Multiple Choice-B	.82	.00
Multiple Choice-C	.85	.00
Free Response-A	.75	.00
Free Response-B	.77	.00
Free Response-C	.82	.00
Constrained Free-Response	.00	.79

Note. All loadings are significant at $p < .001$ level (t range for sample 1 = 14.01 to 35.91; t range for sample 2 = 15.16 to 40.27). Sample 1 completed the 1-bug variants. Sample 2 completed the 3-bug variants.

Appendix

Examples of Multiple-Choice, Free-Response, and Faulty-Solutions Items

6. What output is produced by the following program?

```
program ABC (input, output);
  var
      n : integer;

  procedure Increment (var a, b : integer);
  begin
    a := a + 1;
    b := b + 1
  end;

begin
  n := 3;
  Increment(n,n);
  write(n)
end.
```

(A) 5
(B) 4
(C) 3
(D) 0
(E) An error message

7. Which of the following is (are) true of a compiler?

   I. It is a program that takes object code as input and executes that code.
   II. It is a program that takes source code as input and outputs object code.
   III. It can be written in a language other than the language it will compile.

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

2. Write a Pascal program that simulates a pocket calculator by evaluating, from left to right, an expression consisting of integer constants separated by the operators +, −, *, and / and terminated by a semicolon. For example, given the input

   10 + 2 * 3 − 4

   the program should produce the output 32, which equals ((10+2)*3)−4 and NOT 10+(2*3)−4. Blanks may occur anywhere in the expression other than within integer constants; if they do occur, they should be ignored by your program.

Rotate Array Program

Program specification: A procedure is needed that rotates the elements of an array s with n elements so that when the rotation is completed, the old value of s[1] will be in s[2], the old value of s[2] will be in s[3],..., the old value of s[n - 1] will be in s[n], and the old value of s[n] will be in s[1]. The procedure should have s and n as parameters. It should declare the type Item and have s be of type List which should be declared as List = array[1..Max] of Item.

Instructions. On the next page is a PASCAL program that was written to conform to this specification. The program contains 1 to 3 bugs (errors). All of the bugs are located within the lines that are triple spaced. The bugs are not syntactic; the program will compile and execute, but it will not produce the desired results. On the program on the right, correct the bugs by deleting lines and/or inserting new ones. Use the program on the left as your reference copy (both programs are exactly the same). The insertions and deletions you make will be recorded on a carbon copy of the program that you may keep. To keep the copy legible, use scratch paper to work out the exact form of the code you wish to insert, and erase only when absolutely necessary.

To delete a line, place a D in the space before it and draw a line through the code like this:

D ~~s[i] := s[i - 1];~~

To insert a new line, write in the new code and then place an I in the space to the left of it. For example:

I s[i] := s[i + 1];

Do not use arrows to indicate where lines should be moved in the program; use the delete-and-insert technique instead. If you want to change part of a line, you should delete the whole line and insert the corrected one.

Remember to write your name, date of birth, and school at the top of each sheet and to print legibly.

YOU SHOULD TAKE NO LONGER THAN 20 MINUTES TO COMPLETE THIS PROBLEM.