PREDICTING PERFORMANCE IN AN INTRODUCTORY COMPUTER SCIENCE COURSE

A group of 269 first-semester freshmen was used to predict both performance in an introductory computer science course and first-semester college grade point average by using information regarding the students' programs and performance in high school along with American College Testing Program (ACT) test scores.

D. F. BUTCHER and W. A. MUTH

As the number of students wishing to major in computer science has increased, colleges and universities have been forced to restrict the number of majors because of shortages of faculty, staff, and computing facilities. The typical practice has been to restrict enrollment on the basis of a student's high school grade point average and/or performance on a standardized exam such as the ACT or SAT exam.

Prior to August 1983, any student eligible for admission to West Virginia University was permitted to select pre-computer science as an area of study for the first two years. The number of students subsequently permitted to enter the computer science major was controlled by performance in four computer science courses to be completed during the first two years of study.[1]

Now that the computer community (IEEE Computer Society and the Association for Computing Machinery) is considering accreditation of computer science degree programs and preliminary guidelines indicate that it will be necessary to restrict enrollment, there must be concern about developing academically justifiable procedures for restricting enrollment in these accredited programs. It was decided to investigate the feasibility of predicting performance for pre-computer science majors by using information available prior to their enrolling in college.

The current study appears to be unique in that only high school data and ACT scores were used to predict college performance in computer science courses that contained only pre-computer science students.

REVIEW OF LITERATURE

Most recently, Campbell and McCabe [3] considered sex of the student, SAT scores, and high school grades in order to develop a linear discriminant function. They were able to classify 175 of 256 students (68.4 percent) into the correct group: either into "CS+" defined as those students who had successfully completed one year of study in computer science, engineering, or other science, or into "other," which was all other majors.

Alspaugh [1] used a coding for mathematical background and scores on (1) the Thurstone Temperament Schedule, (2) the IBM Programmer's Aptitude Test, (3) the Watson-Glaser Critical Thinking Appraisal, and (4) the SCAT Quantitative and Verbal Subtests to predict performance in a computer programming course. Fifty students were available for the study, and three measures of programming proficiency were used: an assembly language score, a FORTRAN score, and the total of the two scores. Alspaugh was able to explain from 33 to 40 percent of the variation in programming proficiency using 9 or 10 independent variables selected from a total of 18 independent variables. For all 3 dependent variables, a mathematics background code that attempted to interpret level of high school and college mathematics appeared to be the most important independent variable.

[1] The four courses, to be taken in sequence, are CS 1, 2, 50, and 51. CS 1, 2, and 51 cover higher level language programming through file organizations, list processing, and data structures. CS 50 covers assembler language programming. A student must attain a 2.5 grade point average (4.0 = A) in CS 1 and 2, and then a 2.5 grade point average in CS 50 and 51 in order to enter the computer science major. Students are allowed a total of three registrations, if necessary, for each of the two-course sequences; withdrawal from a course counts as one registration. These requirements remain in effect with the added requirement that two semesters of calculus be completed with at least C grades in each.

Konvalina, Stephens, and Wileman [6] concluded that high school performance, high school mathematics background, and exposure to high school computer course work were related to performance in college-level computer science courses. They used 165 students enrolled in an introductory computer science course and were able to account for about 19 percent of the variation in final examination score.

Stevens [11] randomly selected 73 students from a computer course and administered a Group Embedded Figures Test to classify students into field-independent or field-dependent groups. The 25 students able to identify the largest number of figures were classified as field independent, whereas the 21 students who identified the smallest number of figures were classified as field dependent. She determined an achievement score by using the sum of two subscores: a general computer literacy and computer educational usage score, and a computer knowledge and BASIC programming score. She concluded that there were significant differences in achievement score means between field independent–dependent students, with field-independent students having the highest achievement test scores.

Stephens, Wileman, and Konvalina [12] reported a simple correlation of $r = 0.47$ between their Computer Science Aptitude Pretest and final exam scores for 183 students enrolled in a course that contained programming in PL/C and simulated assembler language.

Fowler and Glorfeld [5] developed a logistic classification model that used college grade point average (GPA), number of math courses, SAT math score, and age as independent variables in an attempt to classify students as "high aptitude" (student received an A or B grade in an introductory programming course) or "low aptitude" (received C, D, or F). The model correctly classified 122 of 151 students (81 percent); a subsequent validation of the model correctly classified 41 of 55 students (79 percent). Their results indicated that college GPA was the most important independent variable, whereas age was of marginal value.

Mazlack [7] concluded that "future programming skill is not predictable" by the IBM Programmer's Aptitude Test (PAT) based on results of his study of 63 students enrolled in a FORTRAN programming course. The simple correlation between total PAT score and course grade would only explain about 11 percent of the variation in course grade.

Peterson and Howe [8] used biographical temperament and aptitude data on 113 students enrolled in an introductory computer science course during the fall of 1975 to build a mathematical model, and 119 students enrolled during the spring of 1976 to validate their model. The course covered general information, programming, and computers and society. College GPA and General Intelligence score were found to be important in predicting the grade received in the course.

DESCRIPTION OF THE DATA

Enrollment in pre-computer science at West Virginia University was not restricted until the fall of 1983. Therefore, these data represent an unselected sample of first-semester freshmen students who completed the introductory computer science course for computer science majors (CS 1) during the fall of 1981 and the fall of 1982. Only first-semester freshmen were used since students who had completed college credits would have had an opportunity to remove deficiencies indicated by their ACT scores and high school transcripts.

Examination average (EXAM), laboratory average (LAB), and final course grade (GRADE; A = 4) were used to measure success in the introductory computer science course, whereas first-semester college grade point average (CGPA) was used to measure overall success in college.

Thirteen variables were used as independent variables. ACT scores for mathematics (ACT-M), English (ACT-E), natural science (ACT-NS), social science (ACT-SS), and composite (ACT-C) were obtained from standardized ACT examination reports. The ACT-C score is a linear function of the other ACT scores and the student's "self-reported" high school grade point average. In other words, inclusion of all five ACT scores does not create a linear dependency.

The student's class rank (HS-RANK), class size (HS-SIZE), high school grade point average (HS-GPA), level of high school mathematics (HS-MATH), whether or not the student completed a computer programming or data processing course in high school (HS-CS), number of physics and chemistry courses completed in high school (HS-PC), and number of mathematics, physics, chemistry, biology, and computer courses completed in high school (HS-SCI) were obtained from high school transcripts. Finally, the student's percentile rank (HS-PER) was computed as HS-PER = (100 − ((HS-RANK/HS-SIZE) * 100)).

HS-MATH was coded as 0, 1, or 2 depending upon the number of mathematics courses completed from Algebra II and precalculus mathematics. HS-CS was coded 0 for students who did not complete a computer course in high school and 1 for students who did. HS-PC was coded 0, 1, or 2 depending on whether the student completed 0, 1, or 2 (or more) physics and/or chemistry courses in high school.

To measure the student's overall preparation for college, HS-SCI was coded as the total number of mathematics, physics, chemistry, biology, and computer courses completed in high school.

Sex of the student was not used in the analysis since sex cannot be used as an admission criterion. Also, it was felt that some of the effect frequently associated with sex differences could be explained by variables that measure the student's preparation for college.

Variations due to instructors and any year-to-year differences were not considered since the objective was to establish criteria for admission of future students into the pre-computer science program of study.

STATISTICAL METHODS

SAS, the Statistical Analysis System [10], was used to perform all statistical analyses.

Analyses of variance were used to partition the among group sum of squares into that due to linear relationship with 1 degree of freedom (d.f.) and that due to lack of fit with $k - 2$ d.f. for HS-MATH, HS-PC, and HS-SCI. This was not necessary for HS-CS because a linear equation will pass through the two points. In other words, the d.f. for lack of fit $(k - 2)$ is zero for HS-CS because there is no lack of fit in this case. For a detailed discussion of this topic, the reader is referred to Walpole and Myers [13, pp. 298–303] or Draper and Smith [4, pp. 33–42].

The "best" equation for predicting each dependent variable was found by looking at all possible equations that could be formed with 13 independent variables. The "best" k-variable equation was defined as the equation with the highest multiple correlation coefficient squared (R^2) in which all variables are significant ($p < .05$).

RESULTS AND DISCUSSION

A total of 372 first-semester freshmen registered for CS 1 during the fall semester for the two years in the study. Of these 372 students, 64 were deleted from the study because they withdrew from the course and 2 additional students were deleted because they were unable to complete the course due to illness. To have included these 64 in the study would have required the assigning of some grade for the course—only W (withdrew) and F (failing) were candidate grades, and the authors deemed each to be inappropriate. Finally, 37 students were deleted because of inability to locate complete high school data for them. Thus, data on 269 first-semester pre-computer science freshmen who completed CS 1 were used: 124 from the fall of 1981 and 145 from the fall of 1982.

Table I gives the means, standard deviations, and minimum and maximum values for all variables used in the study. The mean of 1.67 for HS-MATH indicates that most students completed both Algebra II and precalculus mathematics. It is seen also that 40 percent of the students completed a computer programming or data processing course in high school, as HS-CS = 0.40.

Because HS-MATH, HS-CS, HS-PC, and HS-SCI were created to summarize high school transcripts, one-way analyses of variance were used to determine whether a linear relationship between each dependent and independent variable would adequately explain variability among dependent variable means. That is, the sum of squares among groups formed by each independent variable was partitioned into that due to a linear relationship and that due to lack of fit to determine if the linear model adequately fit the data.

Table II gives the dependent variable means by level of high school mathematics completed (HS-MATH); for all four dependent variables, differences among means were significant ($p < .01$); the linear term was significant ($p < .01$); the lack-of-fit term was not significant ($p > .10$). As expected, mathematics course work in high school improved student performance in college. Those students who completed both Algebra II and precalculus mathematics performed better than those with only Algebra II, whereas those with Algebra II performed better than those who did not complete either course in high school for all four measures of success.

Table II further gives the dependent variable means

TABLE I. Means, Standard Deviations, and Minimum and Maximum Values for First-Semester Freshmen Used in the Study ($n = 269$)

Variable	Mean ($\bar{X}$)	Standard deviation	Minimum	Maximum
EXAM	72.8	15.4	13.5	95.8
LAB	78.5	21.7	0.0	100.0
GRADE	2.33	1.29	0	4
CGPA	2.60	0.88	0.000	4.000
ACT-M	22.8	5.9	2	35
ACT-E	20.1	4.9	4	33
ACT-NS	24.4	5.8	4	35
ACT-SS	20.2	6.9	2	34
ACT-C	22.0	5.0	5	33
HS-RANK	84.2	102.9	1	668
HS-SIZE	311.0	199.5	15	1062
HS-PER	75.5	21.0	16	100
HS-GPA	3.23	0.52	2.03	4.000
HS-MATH	1.67	0.59	0	2
HS-CS	0.40	0.49	0	1
HS-PC	1.28	0.75	0	2
HS-SCI	5.53	1.48	1	9

$\bar{X} = (\sum_{i=1}^{n} X_i)/n$
$s^2 = (\sum_{i=1}^{n} (X_i - \bar{X})^2)/(n - 1)$
$s = \sqrt{s^2}$

TABLE II. Means of Dependent Variables by HS-MATH, HS-CS, HS-PC, and HS-SCI

Independent variable	N	Dependent variable			
		EXAM	LAB	GRADE	CGPA
HS-MATH					
None (= 0)	17	56.3	62.3	1.22	1.66
Algebra (= 1)	54	67.7	72.3	1.81	2.24
Precalculus (= 2)	198	75.6	81.6	2.56	2.77
HS-CS					
No (= 0)	161	72.1	79.1	2.31	2.57
Yes (= 1)	108	73.9	77.6	2.35	2.64
HS-PC					
None (= 0)	48	63.5	67.8	1.58	2.09
One (= 1)	96	73.4	76.3	2.29	2.59
Two or More (= 2)	125	75.9	84.3	2.64	2.80
HS-SCI					
$k = 1$	3	53.9	62.9	1.33	1.53
$k = 2$	8	68.1	71.6	2.00	2.37
$k = 3$	15	60.1	54.1	1.13	1.74
$k = 4$	30	69.1	74.1	2.00	2.25
$k = 5$	60	67.1	76.0	1.97	2.44
$k = 6$	84	78.7	85.0	2.81	2.90
$k = 7$	55	74.8	81.4	2.47	2.76
$k = 8$	11	82.5	80.3	2.73	2.83
$k = 9$	3	81.9	86.0	3.00	2.93

for those who did and did not complete a computer programming and/or data processing course in high school (HS-CS). For these analyses, differences between means were not significant ($p > .10$) so that the presence or absence of a high school computer course had no effect on performance in computer science or overall performance during the first semester in college.

Differences among dependent variable means were significant ($p < .01$) for students who completed zero, one, and two or more high school physics and chemistry courses (HS-PC) (see Table II). In all cases the linear effect was significant ($p < .01$) and the lack-of-fit (or quadratic given linear) term was not significant. However, the lack-of-fit term was approaching significance for EXAM ($p = .058$).

Freshmen enrolled in this computer science course completed from one to nine mathematics, physics, chemistry, computer science, and/or biology courses in high school. Table II gives the dependent variable means by number of science courses completed in high school (HS-SCI). Although some means are based on very few observations and there is fluctuation among means, a positive linear trend is apparent. Differences among means were significant ($p < .01$), and the linear term was by far the most important term ($p < .01$) for each dependent variable. However, the lack-of-fit term was significant for EXAM ($p = .013$), LAB ($p = .032$), and GRADE ($p = .022$) and was approaching significance ($p = 0.053$) for CGPA. In an attempt to explain the lack-of-fit term, tests for quadratic given linear ($p > .05$) and cubic given linear and quadratic ($p > .05$) effects were not significant. Therefore, even though the lack-of-fit term is significant, fourth or higher degree polynomial equations would be very difficult to explain and were not fit in this analysis.

Simple product moment correlations (r_{xy}) between all four dependent variables (y) and all independent variables (x) are given in Table III. Even though most of these correlations are significantly different from zero, the value of using most of these independent variables to predict the dependent variables is very limited. If one were to square any value r_{xy} in Table III, one obtains a value that is analogous to R^2 in multiple regression—each is a measure of variability in the dependent variable associated with the independent variables. As an illustration, the r_{xy}^2 value between HS-PER and EXAM is $(0.402)^2 = 0.16$, meaning that one could explain 16 percent of the variation in EXAM due to HS-PER.

From these simple correlations, it can be seen that ACT-M has the highest linear relationship with EXAM and GRADE, whereas HS-GPA has the highest simple correlation with the two remaining dependent variables, LAB and CGPA.

Since linear terms explained most of the association between dependent and independent variables, all possible one-, two-, and three-variable regression equations were computed to find the "best" equation for predicting each dependent variable.

ACT-M and HS-GPA produced the "best" two-variable equations for all four dependent variables, and once these two variables were in the equation, no other variable contributed a significant ($\alpha = 0.05$) amount to the ability to predict any of the dependent variables. Other results were not as consistent, but HS-GPA and ACT-C ranked second for CGPA, third for GRADE, and fourth for LAB.

As GRADE and CGPA are accepted measures of overall success, additional analyses were conducted using the three two-variable equations that could be created from HS-GPA, ACT-M, and ACT-C. Table IV gives the results of these analyses. All three equations do a rea-

TABLE III. Product Moment Correlation Coefficients (r_{xy}) between Dependent and Independent Variables Using 269 First-Semester Freshmen

Independent variables	Dependent variable			
	EXAM	LAB	GRADE	CGPA
ACT-M	0.580*	0.378*	0.519*	0.482*
ACT-E	0.470*	0.328*	0.439*	0.533*
ACT-NS	0.416*	0.229*	0.367*	0.351*
ACT-SS	0.355*	0.261*	0.357*	0.382*
ACT-C	0.526*	0.351*	0.490*	0.504*
HS-RANK	−0.278*	−0.207*	−0.311*	−0.407*
HS-SIZE	−0.015	0.011	−0.055	−0.076
HS-PER	0.402*	0.388*	0.442*	0.545*
HS-GPA	0.478*	0.451*	0.510*	0.600*
HS-MATH	0.344*	0.257*	0.307*	0.348*
HS-CS	0.060	−0.035	0.016	0.037
HS-PC	0.269*	0.283*	0.287*	0.280*
HS-SCI	0.324*	0.258*	0.279*	0.314*

* Correlations with absolute values greater than 0.12 are significantly different from zero at $\alpha = 0.05$ level of significance.

TABLE IV. Regression Equations and Squared Multiple Correlation Coefficients for Predicting Dependent Variables*

Equation	Regression coefficients for predicting			
	EXAM	LAB	GRADE	CGPA
Intercept (b_o)	19.84	12.77	−2.255	−0.900
HS-GPA (b_i)	8.025	14.689	0.860	0.807
ACT-M (b_j)	1.188	0.805	0.079	0.039
R^2	0.395	0.242	0.366	0.417
Intercept (b_o)	20.75	13.97	−2.203	−0.911
HS-GPA (b_i)	8.238	15.349	0.866	0.777
ACT-C (b_j)	1.155	0.683	0.079	0.046
R^2	0.333	0.222	0.329	0.410
Intercept (b_o)	35.20	43.21	−0.595	0.606
ACT-M (b_i)	1.155	0.995	0.077	0.032
ACT-C (b_j)	0.513	0.573	0.053	0.057
R^2	0.346	0.149	0.285	0.271

* To find the predicted exam average for a student who had a HS-GPA of 3.50 and an ACT-MATH score of 27, compute

$$\hat{y} = b_o + b_iX_i + b_jX_j$$

$$= 19.84 + 8.025(3.50) + 1.188(27) = 80.0.$$

R^2 measures the fraction of total variability explained by the regression equation.
$R = \sqrt{R^2} = r_{y\hat{y}}$; the correlation between the observed and predicted values.

sonable job of predicting GRADE and EXAM, whereas the two equations that contain HS-GPA do a reasonable job of predicting CGPA and LAB. However, inability to predict LAB is apparent for all three equations.

In order to be admitted to pre-computer science as a freshman, current departmental admission requirements state that an applicant must satisfy at least two of three requirements: (1) have a HS-GPA of at least 3.30, (2) have an ACT-M score of at least 24, and (3) have an ACT-C score of at least 24. Table V contrasts the number of students in this study who would satisfy each of the three admission requirements with the grade earned in the course. The percentage of students earning an A or B in the course is much higher for those who satisfy a current specific admission criterion. Nonetheless it is important to note that a large percentage of those who would not satisfy that current admission criterion still earned an A or B in the course.

Students were further classified into two groups: those who would have been admitted currently to pre-computer science as freshmen and those who would not (i.e., in accordance with the requirements listed in the previous paragraph). Table VI tabulates the final course grade for each of the two groups. Only 123 of the 269 (46 percent) students would satisfy current admission requirements, and the vast majority (75 percent) of these students earned an A or B in the course. However, a reasonable percentage (48 of 146; 33 percent) of those who would not satisfy current admission requirements also earned an A or B.

Overall success should be measured by whether a student was or was not subsequently accepted from pre-computer science into the computer science degree program. As there is no time limit on eventual acceptance but there is a limit on the number of courses a student can repeat, those students who had not satisfied admission requirements within 24 months after enrolling in CS 1 were placed in the "not admitted" group. Table VII compares the number of students admitted to the degree program with those satisfying current pre-computer science admission requirements as freshmen. Only 68 of 269 students (25 percent) were admitted to the computer science degree program. Of these 68 admitted, 20 would not currently satisfy current freshmen admission requirements to enter pre-computer science. One can assume that, of the 75 students who would satisfy current admission requirements, most would have been accepted into the computer science degree program had they decided to pursue a career in computer science, and the 126 who were not accepted into the degree program and would not satisfy pre-computer science admission requirements would likely have difficulty with courses. That is, in one case the students made the choice, whereas in the other the choice was forced upon the students.

SUMMARY AND CONCLUSIONS

The ACT Technical Report [2, pp. 136–138] summarizes results from 437 colleges during 1965–1967 and 419 colleges during 1968–1970, where ACT-M,

TABLE V. Number of Students Who Satisfied Current Admission Requirements for HS-GPA, ACT-M, and ACT-C by Grade Earned in Introductory Computer Science Course

Grade	HS-GPA		ACT-M		ACT-C		Totals
	Yes	No	Yes	No	Yes	No	
A or B	95	45	100	40	82	58	140
C, D, or F	35	94	43	86	34	95	129
Totals	130	139	143	126	116	153	269

TABLE VI. Final Grade in Introductory Computer Science Course for Students Who Would and Would Not Satisfy Current Pre-Computer Science Admission Requirements

Satisfy current admission requirements	Final grade					Totals
	A	B	C	D	F	
Would	44	48	18	8	5	123
Would not	10	38	40	28	30	146
Totals	54	86	58	36	35	269

TABLE VII. Number of Students Admitted to Computer Science Degree Program by Whether They Satisfy Current Pre-Computer Science Admission Requirements as Freshmen

Qualify for admission as freshmen	Admitted to degree program		Totals
	Yes	No	
Would	48	75	123
Would not	20	126	146
Totals	68	201	269

ACT-E, ACT-NS, and ACT-SS were used to predict first-semester GPA. The median multiple correlation for the 437 colleges during 1965–1967 was $R = 0.479$, whereas the median multiple correlation for the 419 colleges during 1968–1970 was $R = 0.465$. In Table III of the present study, the simple correlation (r) between CGPA and each of ACT-M, ACT-E, and ACT-C exceeds the median multiple correlation reported in the ACT Technical Report. Therefore, results from the present study compare favorably with results of other studies. A multiple correlation from an equation using more than one independent variable must at least equal the largest of the simple correlations between each dependent and independent variable.

Results of this study indicate that it is possible to predict performance in an introductory computer science course based on information available from high school transcripts and standardized (ACT) exam scores.

Even though it is possible to predict performance in an introductory computer science course using any two of HS-GPA, ACT-M, and ACT-C, only 36.6 percent of the variation in GRADE ($R^2 * 100$; Table IV) can be accounted for by this relationship. This means that, on the average, as HS-GPA, ACT-M, or ACT-C increases, the probability that a student will succeed in introductory computer science increases; still, most of the variation in EXAM remains unexplained (63.4 percent).

Only data available before the students enrolled in college were used as independent variables. However, as most students are admitted to college during their senior year of high school, the admissions officer could not use the regression equations to predict the student's computer science grade until the student had completed high school.

Correlations with LAB tend to be lower than those for EXAM, GRADE, and CGPA. As LAB includes both in-class quizzes and homework assignments, this quite likely reflects availability of assistance on homework assignments. Also, perhaps those students having trouble with quizzes and examinations spend a disproportionate amount of time on homework to try to earn a passing grade in the course.

It is surprising that exposure to a computer course in high school did not influence performance in the course or performance during the first semester of college. Even though one of the objectives of CS 1 is to teach structured PL/1 programming and no student was exposed to PL/1 in high school, one would expect knowledge of any high-level programming language would be beneficial. It may be that all of these students would have completed a computer course in high school if it had been available, or some may have learned a programming language without a formal course. Furthermore, criteria for high school computer science courses are not published, so it may be that some high school courses did not require students to learn a programming language. Perhaps this documents failure of computer science departments to assist secondary education with the development of meaningful high school computer science courses.

HS-GPA and ACT-M score appear to be the best predictors of performance in computer science and performance during the first semester of college, but an equation based on HS-GPA and ACT-C score also provides a reasonable predictor of college performance. Although the combination of ACT-M and ACT-C scores is acceptable, several of the other possible combinations would do a better job of predicting both the grade in computer science and first-semester CGPA.

Comparisons between the grade earned in the course and whether the student would satisfy current admission requirements yield a mixture of "good news" and "bad news" that gives pause for reflection. It is reassuring that those who would have satisfied present admission requirements did well in the course, yet troubling that one-third of the students who would not have satisfied admission requirements also did well in the course. Thus, one must honestly face up to the need for departmental administrators to exercise prudence and caution; if one has a mechanism for the later acceptance of some students as "transfers" after the student has displayed an ability to perform well in college-level studies, then one may feel somewhat more secure in using the restrictive (initial) admission requirements.

As the computer profession implements criteria for accreditation of computer science degree programs, one should remember that

(1) all studies that used standardized examination scores, HS-GPAs, CGPAs, and college grades concluded that college grades or GPAs were important parameters [1, 5, 8];
(2) a student's desire to succeed is an important factor that is hard to measure;
(3) a large number of students with mediocre high school grades and standardized examination scores are able to succeed;
(4) more than 50 percent of the variation in GRADE and CGPA remains unexplained;
(5) some students score poorly on standardized examinations because they did not have access to advanced course work in high school.

If it becomes necessary to restrict enrollment in computer science programs, provisions should be made to accept transfer students who have demonstrated their ability to handle college-level work. Students should have the opportunity to develop their full potential.

REFERENCES

1. Alspaugh, C.A. Identification of some components of computer programming aptitude. *J. Res. Math. Educ.* 3, 2 (Mar. 1972), 89–98.
2. American College Testing Program. Assessing students on the way to college. Tech. Rep. for the ACT Assessment Program, American College Testing Program, Iowa City, Iowa, 1973.
3. Campbell, P.F., and McCabe, G.P. Predicting the success of freshmen in a computer science major. *Commun. ACM 27*, 11 (Nov. 1984), 1108–1113.
4. Draper, N., and Smith, H. *Applied Regression Analysis.* 2nd ed. John Wiley & Sons, New York, 1981.
5. Fowler, G.C., and Glorfeld, L.W. Predicting aptitude in introductory computing: A classification model. *AEDS J. 14*, 2 (Winter 1981), 96–109.
6. Konvalina, J., Stephens, L., and Wileman, S. Identifying factors influencing computer science aptitude and achievement. *AEDS J. 16*, 2 (Winter 1983), 106–112.
7. Mazlack, L.J. Identifying potential to acquire programming skill. *Commun. ACM 23*, 1 (Jan. 1980), 14–17.
8. Peterson, C.G., and Howe, T.G. Predicting academic success in introduction to computers. *AEDS J. 12*, 4 (Summer 1979), 182–191.
9. Ralston, A., and Shaw, M. Curriculum 78—Is computer science really that unmathematical? *Commun. ACM 23*, 2 (Feb. 1980), 67–70.
10. SAS Institute. *SAS User's Guide: Statistics.* 1982 ed. SAS Institute, Cary, N.C., 1982.
11. Stevens, D.J. Cognitive processes and success of students in instructional computer courses. *AEDS J. 16*, 4 (Summer 1983), 228–233.
12. Stephens, L.J., Wileman, S., and Konvalina, J. Group differences in computer science aptitude. *AEDS J. 14*, 2 (Winter 1981), 84–95.
13. Walpole, R.E., and Myers, R.H. *Probability and Statistics for Engineers and Scientists.* 2nd ed. MacMillan Publishing Co., New York, 1978.

CR Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education—*self-assessment*; K.3.m [**Computers and Education**]: Miscellaneous—*accreditation*
General Terms: Human Factors, Measurement
Additional Key Words and Phrases: computer science admission requirements, predicting performance in computer science using ACT scores, predicting performance in computer science using high school transcripts

Authors' Present Address: D.F. Butcher and W.A. Muth, West Virginia University, Dept. of Statistics and Computer Science, Knapp Hall, Morgantown, WV 26506.