PREDICTING STUDENT SUCCESS IN AN INTRODUCTORY PROGRAMMING COURSE

by Terry R. Hostetler

Global Analytics, Inc.
10065 Old Grove Road
San Diego, California 92131

Abstract

This paper examines to what extent a student's aptitude in computer programming may be predicted through measuring certain cognitive skills, personality traits and past academic achievement. The primary purpose of this study was to build a practical and reliable model for predicting success in programming, with hopes of better counseling students. Results from correlating predictor variables with a student's final numerical score confirmed past studies which showed the diagramming and reasoning tests of the Computer Programmer Aptitude Battery and a student's GPA to be the predictors most closely associated with success. A multiple regression equation developed from 5 predictors correctly classified 61 of 79 students (77.2%) into low and high aptitude groups.

1 Introduction

Instruction in computer science has become increasingly desirable due to several expanding areas of demand. Many academic departments have realized the importance of computer assistance to their discipline, and so have adopted computer science requirements. The public has recognized a growing practical need to be computer literate in order to function as intelligent consumers and take advantage of today's technology. Employers have begun scrutinizing potential applicants for computer training, causing students to carefully consider as extensive a computer science background as possible. These and other demands have resulted in a recent saturation of computer science courses, including the traditional first course, introductory programming.

Advising students who are making their initial contact with computer science in curriculum decisions has always been difficult. The increased volume of students, however, has made it of even greater importance to devise a useful method for determining individualized counseling. Both students and computer science departments can benefit greatly from a screening technique in which students can compare their interest in programming to their projected aptitude.

At educational institutions, such as the University of Illinois at Urbana-Champaign, where the student population has in general met uniformly high admission's criteria, it might be thought that these students would perform at a consistently high level in an introductory programming course. Those involved in teaching such courses, however, observe that student ability has a remarkably broad range, even within apparently homogeneous groups in a particular college.

Considering this seemingly extensive range of aptitude, it becomes desirable to question whether it is possible to identify traits in an individual which may be used to predict how successful that person will be in an introductory programming course. This paper investigates to what extent certain cognitive skills, personality variables, and past academic achievement can be used to develop such a predictive scheme.

2 Design

2.1 Sample Group

Students used in this study were enrolled in Computer Science 105 at the University of Illinois at Urbana-Champaign during the spring semester of 1982. Computer Science 105, entitled "Introduction to Computers and Their Application to Business and Commerce", is offered every semester and has no prerequisite.

Computer Science 105 covers the basic concepts of structured programming, using an extended version of Fortran supported by the Watfiv compiler. Its major topics include: organization of the computer, data types, variables, arithmetic expressions, assignment statements, input/output, control flow, multidimensional arrays, subprograms, and sorting and searching lists.

Students are evaluated according to their performance on programming assignments (seven to nine), two one-hour examinations, and a three-hour final examination. Programming assignments require students to apply concepts from lecture material to solve stated problems, using an IBM 4341 computer in time-sharing mode. Exam material ranges from objective questions on theory and facts to the actual writing of programs.

A sample of 120 students was randomly selected from approximately 600 students enrolled. This sample was reduced by 17 students who withdrew before the end of the semester. Missing data forced the elimination of another 24 students, leaving a final sample size of 79. This final sample consisted of: 35 males and 44 females; 34 freshmen, 14 sophomores, 18 juniors, 12 seniors, and one graduate student.

2.2 Selection of Data Variables

Two tests from the Computer Programmer Aptitude Battery (CPAB) were chosen as measures for this study. The author, Palormo [Palo74], describes these tests as:

Reasoning--a test of ability to translate ideas and operations from word problems into mathematical notations.

Diagramming--a test of ability to analyze a problem and order the steps for solution in a logical sequence.

Palormo reports estimated reliabilities of 0.88 for the reasoning test and 0.94 for the diagramming test. Validity results show correlations with training success over four studies varying from 0.43 to 0.52 for the reasoning test and from 0.25 to 0.69 for the diagramming test. These two tests recorded the most consistently high correlations of the five tests composing the battery.

In the only other study located in which the CPAB was included as a predictor, Mussio and Wahlstrom [MuWa71] found the diagramming test of the CPAB the single best predictor of course grade, supporting their conclusion that reasoning ability is the single most important qualification for programmers. They feel that the diagramming test closely resembles basic demands that are relevant to programming, and that it appears the logic required to solve the test questions is also important and necessary in a training situation. Similarly, Johnson [John72], in his review of the battery, asserts that the reasoning test of the CPAB represents a task very close to that of programming.

In the area of personality, Weinberg [Wein71] states that there appears to be evidence indicating that critical personality factors can be located and associated with particular tasks, at least to the extent that their possession may render one incapable of performing that task well. Alspaugh [Alsp72] found that the more successful programming student might be expected to have a personality associated with a low level of "impulsiveness" and "sociability" and a relatively high level of "reflectiveness" as measured by the Thurstone Temperament Schedule.

I selected Form A of the Sixteen Personality Factor Questionnaire (16PF) to be administered for this study. Its manual [IPAT79] describes the 16PF as an objectively scored test constructed through basic research in psychology to provide the most complete coverage of personality possible in a brief time. This comprehensive coverage is based on the measurement of 16 functionally independent and psychologically meaningful traits (PF01, PF02, ..., PF16).

The manual reports the overall reliability of factor scores as quite good, even over a four-year period. Validity coefficients are shown to be exceptionally high, meaning the test questions are good measures of personality traits, as these traits are represented in research analysis.

The significance of past academic achievement in predicting programming success has been established for certain variables by several studies. Petersen and Howe [PeHo79], Fowler and Glorfeld [FoGl81], and Bauer, Mehrens, and Vinsonhaler [BaMV68] all reported college grade point average (GPA) as the single best predictor of success in the models they developed. Studies both by Fowler and Glorfeld [FoGl81] and Alspaugh [Alsp72] found a student's mathematical background to be an important contributing element in estimating success.

GPA and math background were both included as variables in my study. Student GPA's were collected from the registrar as of the beginning of the spring semester 1982. A student's math background was measured according to a scale similar to one used in Alspaugh's [Alsp72] study. This scale associates an integer with the most advanced mathematics course a student has passed, with a higher integer indicating a more advanced course.

The primary criterion of success in Computer Science 105 was chosen to be final numerical score. A student's final numerical score is computed as a weighted sum of objectively graded programming assignments and exams. Students are ranked according to these scores, and then final grades are assigned. In the judgment of those teaching the course, the final numerical score is the most consistent and accurate measure available of successful performance.

The cognitive and personality tests were administered during the first two weeks of the semester in two one-hour sessions. Session one included Form A of the 16PF, an untimed test requiring approximately 40 minutes. The reasoning and diagramming tests of the CPAB, with corresponding test times of 20 and 35 minutes, were given in session two. Results of the tests were not made available to the students.

A summary of the 21 independent and dependent variables used in this study, along with their assigned abbreviations and sources, is displayed in Table 1.

Table 1

Data Variables

Variable	Abbreviation	Source
Cognitive:		
CPAB		
—Reasoning	REASON	CPAB
—Diagramming	DIAGR	CPAB
Personality:		
16PF		
—Reserved/Warmhearted	PF01	16PF
—Less Intelligent/ More Intelligent	PF02	16PF
—Affected by Feelings/ Emotionally Stable	PF03	16PF
—Humble/Assertive	PF04	16PF
—Sober/Happy-go-lucky	PF05	16PF
—Expedient/Conscientious	PF06	16PF
—Shy/Venturesome	PF07	16PF
—Tough-minded/Tender-minded	PF08	16PF
—Trusting/Suspicious	PF09	16PF
—Practical/Imaginative	PF10	16PF
—Forthright/Shrewd	PF11	16PF
—Unperturbed/Apprehensive	PF12	16PF
--Conservative/Experimenting	PF13	16PF
—Group Oriented/ Self-sufficient	PF14	16PF
—Undisciplined Self-conflict/ Controlled	PF15	16PF
—Relaxed/Tense	PF16	16PF
Academic:		
College GPA	GPA	Registrar
Math Background	MATH	Questionnaire
Success:		
Final Numerical Score	SCORE	Instructor

3 Results

The sample of students was randomly divided into two groups: Group A (64 students), used in performing a bivariate correlation analysis and in developing a multiple regression equation, and Group B (15 students), used to cross-validate this regression equation.

Pearson product-moment correlation coefficients were generated for Group A to measure the degree to which variation in each independent variable relates to variation in the dependent variable. Correlations were also tested for significance from zero. Correlations for all predictor variables and their associated significance levels are presented in Table 2.

The predictor most highly associated with success (SCORE) was found to be the diagramming test of the CPAB (DIAGR), followed in order by the reasoning test of the CPAB (REASON) and a student's GPA. These measures obtained highly significant correlations of 0.480, 0.406, and 0.367 respectively. No other independent variables were found to significantly correlate with a student's final numerical score.

Multiple regression using stepwise inclusion was performed with all the independent variables in this study. With stepwise inclusion, the variable that accounts for the largest amount of variance unexplained by the variables already in the equation, enters the equation at each step. The results of this analysis are summarized in Table 3.

Table 2

Correlations of Independent Variables with Final Numerical Score (SCORE) (N = 64)

Independent Variable	Correlation Coefficient
REASON	.406**
DIAGR	.480**
PF01	-.035
PF02	.145
PF03	-.121
PF04	-.056
PF05	-.059
PF06	-.088
PF07	-.025
PF08	.126
PF09	.072
PF10	-.068
PF11	.016
PF12	.008
PF13	-.173
PF14	.036
PF15	-.043
PF16	.102
GPA	.367**
MATH	.030

** $p < .01$

Table 3

Stepwise Multiple Regression Analysis (N = 64)

Step	Variable Entered	Multiple R	Simple R
1	DIAGR	.480**	.480**
2	GPA	.568**	.367**
3	REASON	.608**	.406**
4	MATH	.632**	.030
5	PF05	.653**	-.059
6	PF08	.658**	.126
7	PF10	.666**	-.068
8	PF09	.671**	.072
9	PF01	.677**	-.035

** $p < 0.01$

The multiple correlation (multiple R) considering only the best predictor (step 1), DIAGR was increased from 0.480 to 0.568 with the addition of GPA (step 2). This multiple R was further improved to 0.608 by including REASON (step 3). Although MATH was not significantly correlated with the success criterion, its addition to the model (step 4) raised the multiple R to 0.632. Similarly, PF5, the first personality factor added (step 5), improved the multiple R to 0.653, even though it did not directly correlate with success. The addition of the remaining variables to the regression equation resulted in minimal increments to the multiple R.

Table 4 provides the beta weights and R squared for this five-variable model. Beta weights, the standardized regression weights, show the relative contribution of the corresponding predictor variables to the success criterion. The R squared indicates the proportion of variation in the criterion measure explained by the predictors.

Table 4

Statistics for
Five-Variable Regression Model
(N = 64)

Independent Variable	Regression Coefficient	Beta Weights
DIAGR	.593	.385
GPA	6.442	.330
REASON	.663	.260
MATH	2.260	.191
PF05	-.782	-.167
(constant)	21.675	

Multiple correlation = 0.653**; R squared = 0.427

** p<0.01

The five-variable model was cross-validated using the 15 students in Group B. The cross-validation correlation between the predicted final numerical scores and the actual scores was found to be 0.672 (p<0.01). This cross-validation R suggests that the relationships identified by the model hold true for different samples drawn from the same population. In other words, the formulated model does not appear to be sample dependent.

All analysis described in this study was performed using the Statistical Package for the Social Sciences [NieN75].

4 Discussion and Recommendations

The multiple R of 0.653 obtained for the five-variable equation developed in this study is comparable to research by Alspaugh [Alsp72] and Mussio and Wahlstrom [MuWa71] whose models, measuring a similar combination of traits, derived multiple R's of 0.632 and 0.67 respectively. This study's model explained approximately 43% of the variance in the final numerical scores of students, leaving a majority of the variance unaccounted for.

As with Mussio and Wahlstrom's work [MuWa71], the diagramming and reasoning tests of the CPAB were found to be the highest correlating independent variables (0.480 and 0.406). DIAGR and REASON, when summed together, correlated 0.533 (p<0.01) with success in the course. These results support the contention that reasoning is a cognitive skill important to programming.

None of the personality traits measured correlated significantly with success in the course (SCORE). PF05, though, as the fifth variable added in the multiple regression analysis, improved the multiple R from 0.632 to 0.658.

A student's GPA was found to correlate highly significantly with SCORE, supporting past research which reflects the tendency of past academic success to be a good predictor of current academic success. MATH, the fourth variable added during the multiple regression, raised the multiple R from 0.608 to 0.632, despite its low correlation with success (0.030).

The primary goal of this study was to develop a practical and reliable model for predicting success in an introductory programming course, with the major benefit of being able to better counsel students in curriculum decisions.

In view of this goal, Table 5 shows a breakdown of the student sample according to high and low aptitude, based on predicted and actual final numerical scores. High aptitude is defined to be a final numerical score associated with a final letter grade of an A or B, while low aptitude is designated as a score resulting in a C, D, or E.

Table 5

Breakdown of Student Aptitude
for Five-Variable Model

		Actual		
		Low	High	Total
Predicted	Low	30	10	40
	High	8	31	39
	Total	38	41	79

The table shows that 30 out of 40 students (75%) who were predicted to have low aptitude, and 31 out of 39 students (79.5%) who were predicted to have high aptitude, actually did attain those levels of aptitude. Overall, the model correctly classified 61 out of 79 students (77.2%) into low and high aptitude.

Prediction of success based on the models presented in this study is a useful technique in counseling students. A majority of the variance in the success criterion was not explained, however, requiring the consideration of other factors not measured that will aid in predicting a student's success, such as measures of an individual's desire and motivation. Future research in this area must be done in an attempt to identify such traits.

It is important to note that this study dealt with a highly homogeneous, pre-selected group of students. Most members of the sample group were enrolled in the College of Commerce and Business Administration, for which CS105 is a required course. This college reports that its students entering in 1981 had an average ACT composite score of 27 and high school class rank of 92%.

Acknowledgements

I am grateful to Kikumi and Maurice Tatsuoka, Geneva G. Belford, and Gerard M. Chevalez for their help throughout this research. I also extend appreciation to the students of my sample group for participating in this study.

(References given on page 49)

18. Phillipakis, A.S. and Kazmier, L.J. (1982). Advanced COBOL, McGraw-Hill, N.Y.

19. Pressman, R. (1982). Software Engineering: A Practitioners Approach, McGraw-Hill, N.Y.

20. Rindfleisch, D.H. (1976). Debugging System 360/370 Programs Using OS and VS Storage Dumps, Prentice-Hall, Englewood Cliffs, N.J.

21. Shelly, G. and Cashman, T. (1978). Advanced Structured COBOL, Anaheim Press, Fullerton, CA.

22. Tremblay, J. and Bunt, R.B. (1979). An Introduction to Computer Science, McGraw-Hill, N.Y.

23. Tremblay, J. and Sorensen, P.G. (1976). An Introduction to Data Structures with Applications, McGraw-Hill, N.Y.

24. Van Tassel, D. (1978). Program Style, Design, Efficiency, Debugging, and Testing, Prentice-Hall, Englewood Cliffs, N.J.

25. Warnier, J. (1976). Logical Construction of Programs, Van Nostrand-Reinhold Co., N.Y.

26. Weinberg, V. (1980). Structured Analysis, Prentice-Hall, Englewood Cliffs, N.J.

27. Yarmish, R. and Yarmish, J. (1979). Assembler Language Fundamentals, Addison-Wesley Pub. Co., Reading, Mass.

28. Yourdon, E. (1975). Techniques of Program Structure and Design, Prentice-Hall, Englewood Cliffs, N.J.

29. Yourdon, E. and Constantine, L.L. (1978). Structured Design, 2nd ed., Yourdon, Inc., N.Y.

(Continued from page 43)

References

[Alsp72] Alspaugh, Carol Ann. "Identification of Some Components of Computer Programming Aptitude." Journal for Research in Mathematics Education, March 1972, pp. 89-98.

[BaMV68] Bauer, Roger, William A. Mehrens, and John F. Vinsonhaler. "Predicting Performance in a Computer Programming Course." Educational and Psychological Measurement, 28 (1968), pp. 1159-64.

[FoGl81] Fowler, George C. and Louis W. Glorfeld. "Predicting Aptitude in Introductory Computing: A Classification Model." AEDS Journal, Winter 1981, pp. 96-109.

[IPAT79] Institute for Personality and Ability Testing. Administrator's Manual for the 16 PF. Champaign, Ill: IPAT, 1979.

[John72] Johnson, Richard T. Rev. of the Computer Programmer Aptitude Battery. In the Seventh Mental Measurements Yearbook. Highland Park, N. J.: Gryphon Press, 1972.

[MuWa71] Mussio, Jerry J. and Merlin W. Wahlstrom. "Predicting Performance of Programmer Trainees in a Post-High School Setting." Proceedings of the Annual Computer Personnel Research Conference, 1971, pp. 26-45.

[NieN75] Nie, Norman H., et al. Statistical Package for the Social Sciences. 2nd ed. New York: McGraw-Hill, 1975.

[Palo74] Palormo, Jean M. Computer Programmer Aptitude Battery: Examiner's Manual. 2nd ed. Chicago: Science Research Associates, 1974.

[PeHo79] Petersen, Charles G. and Trevor G. Howe. "Predicting Academic Success in Introduction to Computers." AEDS Journal, Fall 1979, pp. 182-91.

[Wein71] Weinberg, Gerald. The Psychology of Computer Programming. New York: Van Nostrand Reinhold, 1971.