Cognitive Predictors of Success in Computer Programmer Training

Stuart J. Jacobs
Southern Connecticut State College

INTRODUCTION

The aptitudes and abilities required for the position of programmer, within the computer industry, have yet to be fully studied and their inter-relationships known. Although the industry is relatively new, a substantial amount of research in the areas of personnel selection, evaluation and job requirements has been undertaken. Yet these studies have confined themselves primarily to the use of interest scales, aptitude and achievement tests as overall predictors for on-the-job success rather than in the study of the cognitive factors pertinent to the tasks of which programming is composed.

In a study by Deutsch and Shea, Inc. (1963), the relationship between the programmer and the computer is seen as analogous to that of the mahout and his elephant. As with the mahout, the programmer uses his intelligence, skills and abilities in the control and guidance of a powerful and flexible, yet non-intelligent, tool in the performance of specific finite operations which contribute to the completion of more complex tasks. It is the programmer who, when presented with a problem from science, engineering or business, must work out a solution and then formulate the detailed instructions, in a specialized machine understandable language, which are used to direct the computer in the completion of the solution. Consequently, the programmer must have an understanding of the computer and its software, the ability to comprehend complex tasks, ideas and problems, and the logical and analytical ability to restructure these tasks and problems into the simple sequential instructions with which the computer is only able to work.

John and Miller (1957) state that all problems have two general parts: the specific components involved (i.e., data, etc.) and the relationships which are the orderings of or changes to the specific components. Furthermore, they go on to say that knowledge of the specific components and the ability to manipulate this knowledge, according to the constraints imposed by the relationships between them, are the two requirements of problem solving. These relationships and their implications are the concern of logic and logical reasoning, of which inference, deduction, interpretation, and recognition and application of basic assumptions are important cognitive components. A study by Berger (1968) bears out the importance of logical reasoning in programmer trainee performance when a correlation of r = .59 was obtained between a test of logical reasoning and grades from an objective midterm examination in an introductory computer programming course.

In the same study Berger also found a high correlation of r = .55 between advanced mathematics and the aforementioned midterm examination grades. This finding would imply that mathematical ability, aptitude, and abstract aptitude would have significant correlations with programmer trainee performance. Bauer, Mehreus, and Vinsonhaler (1968) also found a strong correlation between mathematical aptitude and trainee performance. C. C. Upshall and L. H. Riland (1958), when determining what cognitive factors seemed to be important for programming work, found that mathematical ability was one of the most mentioned factors from interviews with department supervisors and programmers.

STATEMENT OF PROBLEM

Are inferential ability, deductive ability, interpretive ability, general mathematics ability, and the ability to recognize and apply implied rules and basic assumptions related to programmer trainee performance.

HYPOTHESES

It is hypothesized that there is a relationship between:

1. Programmer trainee performance and inferential ability,
2. Programmer trainee performance and deductive ability,
3. Programmer trainee performance and interpretive ability,
4. Programmer trainee performance and general mathematics ability, and
5. Programmer trainee performance and the ability to recognize and apply implied rules and basic assumptions.

DEFINITION OF TERMS

Inferential ability is defined as the ability to infer a logical conclusion from data or premises as measured by part one of the Watson-Glaser Critical Thinking Appraisal.

Deductive ability is defined as the ability to reason from given premises to their necessary conclusions as measured by part three of the Watson-Glaser Critical Thinking Appraisal.

Interpretive ability is defined as the ability to explain and/or interpret given information as measured by part four of the Watson-Glaser Critical Thinking Appraisal.

General mathematics ability is defined as the ability to work with and comprehend mathematics, mathematical concepts, and abstract concepts in general as measured by the mathematics section of the Scholastic Aptitude Test.

The ability to recognize and apply implied rules and basic assumptions is measured by a composite of the scores from part five of the Watson-Glaser Critical Thinking Appraisal and the verbal section of the Scholastic Aptitude Test.

Programmer trainee performance is defined as the ability to understand and correctly utilize the elements of a computer programming language as measured by the grade on an objective final examination.

METHODOLOGY

SUBJECTS

The subjects were the students, both undergraduate and graduate, enrolled in Computer Science 100 and Computer Science 515 at Southern Connecticut State College (SCSC). These one-semester courses are an introduction to computer programming using the RCA DOS SPECTRA 70 FORTRAN IV language.

INSTRUMENTS

1. The Scholastic Aptitude Test was used as a predictor and measure of mathematical aptitude and comprehension. All students had taken the Scholastic Aptitude Test (S.A.T.) prior to admission to undergraduate student study.

2. The Watson-Glaser Critical Thinking Appraisal (W.G.C.T.A.) is a one-hour paper and pencil test of logical and analytical thinking ability. It consists of five subtests: Test I, inference; Test II, recognition of assumptions; Test III, deduction; Test IV, interpretation; Test V, evaluation of arguments. Reliability was

measured by the split-half method with the odd-even coefficients corrected by the Spearman-Brown formula. The reliability coefficients for liberal arts freshmen and seniors were found to be r = .85 and for the separate subtests for the grade 10 normative groups are reported in Table 1. Construct validity as seen by the interrelationships of the subtests and the total test are represented in Table 2. The relatively low subtest intercorrelation coefficients, from .21 to .50, support the contention that relatively distinctive abilities are being measured.

3. An objective one-hour paper and pencil test composed of 25 multiple choice items designed to measure two areas of knowledge; knowledge and use of the FORTRAN IV language and understanding of the methodology and mechanics of computer programming. Reliability was measured by the split-half method splitting the test on an odd-even basis. Reliability for liberal arts students enrolled in the two Computer Science classes, both undergraduate and graduate, was found to be r = .684.

DESIGN

The study consisted of the subtest scores of the W.G.C.T.A. and the S.A.T. as predictor variables and the final examination score as the criterion variable.

PROCEDURE FOR COLLECTING DATA

The experimenter (E) at midsemester time administered the W.G.C.T.A. to all the students enrolled in the C.S. 100 and C.S. 515 classes. The S.A.T. mathematics and verbal scores were provided by the college.

STATISTICAL ANALYSIS

Zero-order and multiple correlations were calculated along with a step-wise multiple linear regression analysis.

RESULTS

The step-wise multiple linear regression was computed on an RCA 70-45 computer located at Southern Connecticut State College, New Haven, Connecticut. The program used was STEPR, a program contained in the IBM Scientific Subroutine Package version 3.

A significance level of 0.1 was selected for all tests of significance.

HYPOTHESIS #1

Hypothesis #1 was that there is a relationship between Programmer trainee performance and inferential ability. Table 4 shows that the zero-order correlation of these two variables is r = -0.215. It can also be seen that this value of r, although negative, is one of the higher values of r.

HYPOTHESIS #2

Hypothesis #2 was that there is a relationship between Programmer trainee performance and deductive ability. Table 4 shows that the zero-order correlation of these two variables is r = 0.2771. It can also be seen that this value of r is one of the higher values of r.

HYPOTHESIS #3

Hypothesis #3 was that there is a relationship between Programmer trainee performance and interpretive ability. Table 4 shows that the zero-order correlation of these two variables is r = 0.1483.

HYPOTHESIS #4

Hypothesis #4 was that there is a relationship between Programmer trainee performance and general mathematics ability. Table 4 shows that the zero-order correlation of these two variables is r = 0.1399.

HYPOTHESIS #5

Hypothesis #5 was that there is a relationship between Programmer trainee performance and the ability to recognize and apply implied rules and basic assumptions. Table 4 shows that the multiple correlation of these two variables is r = 0.3107. This correlation coefficient is between the scores of the final examination and both the S.A.T. verbal section scores and the Watson-Glaser Critical Thinking Appraisal part five scores.

DISCUSSION

CONCLUSIONS

This study attempted to achieve two things. First, an attempt was made to study the aptitudes and abilities pertinent to computer programming. Second, an objective instrument was used to measure programming ability rather than more subjective methods (i.e., interviews, etc.). None of the variables considered in this study was found to be highly intercorrelated, so they are most likely measuring relatively distinctive abilities. The finding of a negative correlation was astonishing and an explanation not known. The independent variables also do not correlate highly with the criterion variable, suggesting that, although there is a fair amount of overlap, they are measuring different abilities than the criterion variable.

The multiple linear regression which accounted for the most variance of the criterion variable accounted for only 18.1% of that variance. Yet this regression of three independent variables (the S.A.T. verbal, part one of the W.G.C.T.A., and part three of the W.G.C.T.A.) to the criterion variable was found to be a significant predictor of the final examination scores. This amounts to saying that even a predictive model only accounts for 18.1% of the variance is valuable. These results emphasize the importance of isolating cognitive factors pertinent to programming and using these factors for predictive purposes.

LIMITATIONS

Experimentation with laboratory control is rarely possible in education because the important variables under consideration cannot be really controlled by the experimenter. The greatest problem is lack of complete randomization.

The subjects of this study were already assigned to the classes in non-random ways. Another uncontrolled variable was student motivation; approximately one-third of the students were taking these courses as required courses while the other two-thirds were primarily enrolled out of curiosity. Furthermore, because of schedule and testing problems only two of the four sections of Computer Science 100 were able to be used in this study. Consequently, the results reported here have limited generalizability, yet show that further in-depth studies should be done in this area.

FURTHER RESEARCH

Further research should extend in two directions. The investigation of pertinent cognitive functions should be continued and the development of behavioral objectives and goals within the fields of computer programmer training and programmer performance should be pursued.

As of this writing, there does not exist an accepted set of behaviorally defined abilities and behavioral objectives which computer programmers must meet. This must be done so that researchers can identify exactly what cognitive functions, both critical and creative, are required to reach those stated behavioral objectives. Until these behavioral objectives and behaviorally defined abilities have been decided upon, very little can be done.

Once behavioral objectives have been defined and related to the tasks of programming and the behaviorally defined abilities required to reach those objectives found, then the cognitive functions associated with the abilities can be studied in respect to the behavioral objectives. Likewise, the development of reliable valid objective evaluation instruments of computer programming will also have to wait for the aforementioned behavioral objectives to be defined.

Therefore, the author recommends that those in the computer field should begin the task of creating an accepted set of behavioral objectives which describe the work involved in computer programming.

BIBLIOGRAPHY

Berger, Raymond M. "Selection of Systems Analysts and Programmer Trainees." Proceedings of the Sixth Annual Computer Personnel Research Conference 1968, pp. 44-63

Deutsch & Shea, Inc. "A Profile of the Programmer." Industrial Relations News, Inc. New York. 1963 pp.2-3

John, E. R. & J. G. Miller. "The Acquisition & Application of Information in the Problem Solving Process." Behavioral Science, II (1957) pp.291-300

Upshall, C. C. and L. H. Riland. An unpublished study at Eastman Kodak Co. 1958 (McNamara, W. T. and J. L. Hughes) "A Review of Research on the Selection of Computer Programmers." Personnel Psychology Vol. 14, 1961, pp. 39-51

TABLE 1

Subtest Split-half Reliability Coefficients for the Grade 10 Normative Groups of the W.G.C.T.A.

Subtest	No. of Items	FORM Ym r_{lI}*
Inference	20	.55
Recognition of assumptions	16	.74
Deduction	25	.53
Interpretation	24	.67
Evaluation of arguments	15	.62

TABLE 2

Intercorrelations of

Subtests for Grade 10

Normative Samples of the

W.G.C.T.A.

Subtest	2	3	4	5	6
1. Inference	.32	.42	.50	.39	.75
2. Recognition of assumptions		.30	.35	.23	.64
3. Deduction			.48	.35	.73
4. Interpretation				.42	.79
5. Evalution of arguments					.62
6. Total test					

TABLE 3

Sums, Means, Variances and

Standard Deviations of All Variables

	SAT V	SAT M	WG 1	WG 2	WG 3	WG 4	WG 5	Final
Sum	23675	22449	494	515	853	828	414	674
Mean	526.1	498.8	10.98	11.44	18.96	18.40	9.20	14.98
Variance	3334	5479	4.34	10.61	8.68	7.24	10.80	15.61
Standard Deviation	57.74	74.02	2.08	3.25	2.94	2.69	3.28	3.95

TABLE 4

Intercorrelations of All Variables

Variable	SAT-V	SAT-M	WGCTA-1	WGCTA-3	WGCTA-4	WGCTA-5
SAT-V	1.0					
SAT-M	0.06342	1.0				
WGCTA-1	-0.0961	-0.0023	1.0			
WGCTA-3	0.0783	0.3681	-0.048	1.0		
WGCTA-4	0.1168	0.2857	0.0907	0.4149	1.0	
WGCTA-5	0.2881	0.0804	0.1500	0.1371	0.2451	1.0
Final	0.2934	0.1399	-0.215	0.2771	0.1483	0.1771

Multiple correlation coefficient of SAT-V and WGCTA-5 to Final Examination

$R_{Final*SAT-V,WGCTA-5} = .3107$

TABLE 5

Analysis of Variance for the Regression of Final Examination Scores from the SAT verbal and Parts One and Three of the WGCTA

Source of Variation	Degrees of Freedom	Sum of Squares	Mean Squares	F value
Regression	3	125.51	41.84	3.06
Residuals	41	561.47	13.69	
Total (corrected)	44	686.8		

$F(.90,3,41) \_ F(.90,3,40) = 2.23$

$p < .05$

TABLE 6

Regression of Final Examination Scores from the SAT verbal and Parts One and Three of the WGCTA

Variable	Regression Coefficient	Std. Error of Reg. Coef.	Computed T Value
SAT verbal (X_1)	0.0176 (B_1)	0.0097	1.8056
WGCTA part 1 (X_2)	-0.3389 (B_2)	0.2693	-1.2589
WGCTA part 3 (X_3)	0.3332 (B_3)	0.1901	1.7524

Intercept is 3.138 (B_0)

Multiple correlation is 0.427

Std. Error of Estimate 3.7

$$R^2 = \frac{SSR}{SST} * 100 = \frac{125.51}{686.98} * 100$$

$$= .181 * 100 = 18.1\%$$

The prediction equation is

$$\hat{Y} = 3.138 + 0.0176X_1 - 0.3389X_2 + 0.3332X_3$$