VALIDATION OF A SELECTION BATTERY FOR COMPUTER PROGRAMMERS

THESIS

Presented to the Graduate Council of the
North Texas State University in Partial
Fulfillment of the Requirements

For the Degree of

MASTERS OF ARTS

By

Michael Tuseth, B. S.
Denton, Texas
August, 1979

Tuseth, Michael, Validation of a Selection Battery for Computer Programmers. Master of Arts (Industrial Psychology), August, 1979, 57 pp., 12 tables, references, 38 titles.

Subjects were 38 computer programmers employed in a national food-retailing corporation. A job analysis provided a basis for criteria development and served to guide the selection of predictors. Ratings of each programmer's job performance by his immediate supervisor, and scores on such tests as the Computer Programmer Aptitude Battery (CPAB), clerical tests, and supervisory judgment test were obtained.

Relationships between tests and criteria were examined to find the best test combination for predicting programming performance. Statistical treatment of data included a principal components analysis of the criteria and a multiple linear regression analysis. A weighted combination of the CPAB Reasoning, a test of clerical ability, and supervisory judgment test was found to be highly correlated with performance ($R = .60$).

TABLE OF CONTENTS

LIST OF TABLES

VALIDATION OF A SELECTION BATTERY FOR COMPUTER PROGRAMMERS

In the last 20 years, the business world has come to rely heavily on the use of the electronic computer. Despite the widespread use of computers in industry, comparatively little research has been conducted to find improved methods of selecting computer programmers.

A need has existed for valid selection instruments that can be used to predict programming success both on the job and in training. As early as 1952, the need for improved selection of computer programmers was recognized (Rowan, 1957). Prior to 1956, computer programmers were usually selected on the basis of weighted application blanks, references, and personnel interviews. Of the few predictors that are used in programmer selection today, most are tests designed by major data processing companies to measure programming aptitude.

A major problem encountered in predicting computer programmer achievement has been the widespread use of general aptitude tests "designed to yield measurements across the total population" (Palormo, 1974, p. 1). As a result, only certain items have been shown to be effective for the specialized population of computer programmers. For this reason, the need for tests of a more specific nature has become apparent.

The Rand Corporation first employed tests in 1952 to improve selection of programmers (Rowan, 1957). Due to test administration problems, tests intended for use with the general population were abandoned in favor of tests thought to be related to programming ability. Nevertheless, when supervisory ratings were used as a criterion, correlation with the tests was found to be "not significantly different from zero" (p. 352).

Since the data processing field is relatively new and dynamic in relation to other occupational areas, a picture of the "typical" computer programmer has not yet had a chance to emerge. Educators and businessmen alike have shared a popular conception/misconception of qualities which characterize the programmer. In the early years of computer development, programming emphasis was on the solving of scientific problems. Data processing courses became associated with mathematics curricula (Alspaugh, 1972). Although some mathematical knowledge is necessary in order to write computer programs, the degree of mathematical sophistication will vary depending on the type of programming performed.

Courses in marketing or finance could easily prove to be more important than algebra or trignometry classes for the programmer who works in business (rather than scientific environment). Through the years, mathematical background has become confused with the concept of mathematical aptitude. Alspaugh (1972) questioned the assumption that a person with mathematical aptitude will also have the aptitude to successfully master computer programming skills.

Studies by IBM (McNamara & Hughes, 1961) revealed that majoring in mathematics was not found to be significantly related to performance as a programmer. Evidence to the contrary was provided in Alspaugh, who found mathematical background to be the largest contributing factor to programming proficiency in an introductory college courses. She concluded, however, that most of the variance was not explained and that "there are obviously other components of aptitude for computer programming yet to be identified" (p. 98).

Researchers generally have agreed that reasoning ability is an important factor relating to an individual's ability to program (Hollenbeck & McNamara, 1965; McNamara & Hughes, 1961; Paloromo, 1974). McNamara and Hughes (1961) have suggested that an apparatus test, the Logical Analysis Device and a paper-and-pencil test, the Programmer Aptitude Test, both measure reasoning ability, in view of their high correlations with each other. Many of the best predictors of programming performance have been the reasoning subtests of IBM's tests of programming aptitude. Further evidence that reasoning ability is an important factor in job performance was provided by Howell, Vincent, and Gay (1967). They found that tests which purport to measure reasoning ability correlated most highly when a simulated work sample was used as a criterion.

The reasoning section of Primary Mental Abilities Test (PMAT) was found to have the greatest weight in regression equations for predicting job and training criteria (Perry, 1964).

Factors other than reasoning ability and mathematical ability have been investigated in connection with programming ability. A programmer should be able to work under pressure and to put in overtime in order to meet deadlines. Hollenbeck and McNamara (1965) wrote:

> If a program is long, it may be subdivided into sections so that a team of programmers can work on each section under the supervision of the most experienced. (p. 102)

Ability to work and cooperate with others may prove to be an important factor contributing to job success. Alspaugh (1972) expected successful programming students to possess a low level of impulsiveness and "sociability" and to rate highly on "reflectiveness" as measured by the Thurstone Temperament Scale.

Requirements for success as a student (where one's work is done on an individual basis) may differ from requirements for successful job performance. On the job, the individual might often be expected to function as a team member. For this reason, but not this reason alone, it has been suggested that predictors which correlate highly with course grades may show a different relationship when job performance is used as a criterion. In his 1969 study, Correnti concluded that his criterion measure "did not measure those qualities considered important in the out-of-college environment of the job" (p. 3719).

According to Hollenbeck and McNamara (1965), the following job factors were most frequently mentioned as being important: ability to think logically, liking for detail, accuracy, ability to work under pressure, ability to work with people, a retentive memory, adjustment to changes, ability to see a problem through, and mathematical ability.

C. H. Rush (in McNamara & Hughes, 1961) in his assessment of a computer programmer, remarked that

> The best programmer . . . is such that he is quite intelligent and his thinking is analytical, imaginative, and flexible. He views each problem as a challenging mental exercise and attacks it from many angles with much enthusiasm. Much of his time is spent in defining the problem well so that he has all the details in mind when he begins flow charting and coding. He is persistent and follows through until his problem is running efficiently. He gives his attention to details but only to be sure he has made allowance for every contingency. (p. 41)

Perry and Cannon (1967) reported that in a study of 1,378 computer programmers, their interests are most similar to optometrists, chemists, engineers, production managers, mathematics-science teachers, and certified public accountants.

However, none of the existing keys on the Revised Strong Interest Blank adequately represented the interests of programmers. They pointed out that a programmer's most striking characteristic is an interest in problem and puzzle-sovling activities. Cronbach (1970) emphasized that programmers must be able to cope with novel problems and write efficient programs in addition to writing routine programs.

Prior to 1961, much of the validity evidence regarding computer programmer selection techniques was supplied in the form of unpublished reports (McNamara & Hughes, 1961). Their review of the literature revealed that many selection techniques were supported by professional opinion only. Several studies lacked such important information as sample size, correlation coefficients, and sex differences in scores.

Denelsky and McKee (1974) have identified two main ways of identifying potential programmers:

1. Creation of special tests designed to measure the specific aptitudes believed to be essential to computer programming,

2. Application of preexisting tests.

They wrote:

> Unfortunatley, the number of published validation studies which have resulted from the use of both types of tests seems disproportionately small in relation to the massive use of these tests as programmer selection devices for government, industry, and specialized training schools. (p. 129)

Tests of computer programming aptitude have been divided into two main types: (a) paper-and-pencil tests, and (b) apparatus tests. The first apparatus test, called the Problem Solving using Information (PSI) apparatus, was developed by John and Miller in 1957. The apparatus was designed specifically to measure problem-solving ability. Another apparatus developed by Bennett and Langmuir which followed the concept of the PSI apparatus was the Logical Analysis Device (LAD). McNamara and Hughes (1961) stated that

> The LAD is used to test problem-solving ability by requiring the subject to light a center indicator light by the manipulation of nine lights in a circular display around the central light. (p. 46)

The Computer Usage Company Programmer Aptitude Test (CUCPAT) requires the examinee to solve logical problems utilizing the lights of a 1401 IBM computer console (Hollenbeck & McNamara, 1965). All apparatus tests, although appearing to be job related, share a number of disadvantages.

1. They require a trained individual who can administer and interpret the test.

2. Only one person can be tested at a time.

3. They are expensive to use.

Guion (1965) has questioned the practical value of the LAD since it correlates highly with more economical paper-and-pencil tests.

Compared to apparatus tests, paper-and-pencil tests have been more widely used in programmer selection. The most widely used and researched test of programming aptitude has been IBM's Aptitude Test for Programmer Personnel (Howell et al., 1967). Originally developed in 1955, it was called the Aptitude Test for EDPM Programmers. It evolved into the Programmer Aptitude Test (PAT) and Revised Programmer Aptitude Test (RPAT) in 1964. It became known as the Aptitude Test for Programming Personnel (ATPP) in 1965.

Oliver and Willis (1963) have reported the PAT to be a valid predictor of grades in a programming course. Alspaugh (1970) found Part I (letter series) and Part III (arithmetic reasoning) correlated significantly with a computer language proficiency measure in a college computer science course.

In a study of computer service operators, Gordon and Dennis (1966) found a correlation of .62 between the Wonderlic Test and the Programmer Aptitude Test. Their results support earlier research by Biamonte (1965) who found a substantial relationship between intelligence as measured by the Wonderlic and programming course grades. Miller (1970) found a significant correlation of .59 between the PAT and grades in a high school vocational data processing course. Howell et al. (1967) reported that Part II (figure analogies) and III (arithmetic reasoning) best predicted a simulated work sample of programming. Data on the RPAT and ATPP have shown them to be "substantially related and that ATPP is as good or better than the RPAT in predicting class grades" (IBM, 1964, p. 7). Since the RPAT has been shown to be a valid predictor of job performance, the test authors suggest that the ATPP will also successfully predict job performance. The 1964 test manual reports a test score/ability to program rank order correlation of .49 (p. 11).

The author of the Aptitude Assessment Battery-Programmers (AABP) has claimed widespread administration of the test to thousands of individuals in companies, institutions, and government agencies (Wolfe, 1970). Little evidence for its validity is available, however. A study by Denelsky and McKee (1974) reported the AABP to significantly predict both training and programming perfomrance.

A recently developed test of programming aptitude, the Computer Programmer Aptitude Battery (CPAB) was introduced by Palormo in 1967. Research data on the CPAB has suggested that it is primarily a measure of reasoning and number ability (Cronbach, 1970). In his review of the test, Veldman (Buros, 1972) has stated that the CPAB "is probably the best device presently available for selection of computer programmers" (p. 1503). Veldman considered the strength of the test to lie in "its close correspondence to the tasks of programming, the clarity of the directions, and the descriptive material concerning criteria and interpretation." He continued that "with such obvious similarity between the various tests and on-the-job behavior, it is surprising that the performance rating validities are not higher."

Palormo (1974) reported correlations ranging from .55 to .86 between the CPAB and RPAT. The CPAB also correlated highly (.50 and .87) with the Thurstone Test of Mental

Alertness for two groups of employed computer programmers. Palormo wrote that "although the CPAB measures the same factors as do these tests to a considerable degree, for most groups it appears to measure some variance not measured by the other tests" (p. 4).

Palormo reported several validation studies of the CPAB. In a 1966 study of trainees at the educational facilities of a major computer manufacturer, the total CPAB score was shown to be a consistently good predictor of success in training. Correlations between the test and grades for three small groups of trainees were .52, .56, and .71, all significant at or beyond the .01 level of confidence.

The total CPAB was shown to be significantly related to previous training grades ($p < .01$) for a group of currently employed programmers in an Eastern utility in 1967. A West Coast research organization in 1965 and 1966 included the CPAB subtests of Number Ability, Reasoning, and Diagramming, in a broad range of test predictors for the selection of programmer trainees. Test/ability to program correlations for all three subtests were significant ($p < .01$). Results were shown to hold up in a cross-validation sample.

CPAB scores for Canadian Armed Forces officers entering a computer programming training course were correlated with course grades. A significant relationship ($p < .01$) was reported between the CPAB total score and the criterion.

The CPAB manual also reported several studies in which on-the-job performance criteria were used instead of training outcome. A 1967 concurrent validity study in a Midwest retail organization yielded a correlation coefficient of .31 ($p < .05$) between total CPAB scores and supervisory ratings of job performance. Martin (1971) found validities ranging from .02 to .36 between the CPAB and supervisory performance ratings. Only one out of four validation groups showed a correlation significant at or beyond the .01 level. In 1972-73 a study of the CPAB's relationship to programming performance was conducted in a diversified paper products company. The CPAB total test and subtests correlated significantly with supervisory ratings of programmers trainee performance. All subtests were significant at the .01 level except for Letter Series which was significant at the .05 level.

Paper-and-pencil tests have economic and administration advantage when compared to the apparatus tests. However, they demonstrate certain weaknesses also. Wolfe (1971) has pointed out limitations of aptitude tests intended for use with programmers. Factors such as "testwiseness" and reading ability, although relating to test performance, may have little or no relationship to job performance. Wolfe pointed out that certain tests such as the Programmer Aptitude Test are "more heavily loaded with mathematical applications than is necessary for business programming" (p. 270). The nature of speeded tests may eliminate individuals whose slow, careful work habits are an asset to job effectiveness, according to Wolfe. He had claimed that aptitude tests which usually consist of numerous, short problems are not capable of evaluating "the applicants' aptitude for prolonged concentration on a long sequence of steps" (p. 272).

Predictors intended for use with the general population also have been widely used in a effort to predict programming performance. Research had been conducted utilizing tests of interests (Bauer, Mehrens, & Vinsonhaler, 1968), personality (Alspaugh, 1972; IBM, 1964), general aptitude and ability (Bauer et al., 1968; Howell, et al., 1967; Katz, 1962) and biographical data (Alspaugh, 1972; Bauer et al., 1968; Howell et al., 1967). Tests such as the Color Naming Test (Howell et al., 1967) and the Brown-Carlsen Listening Comprehension (Upshall & Riland, 1958) have been used with varying degrees of success in predicting programmer performance.

Validation efforts have yielded disappointing results in many cases for reasons other than choice of predictors. In many cases, a relationship between variables has not been found because of the weakness in the criterion measure.

Palormo (1974) cited a validation study of the CPAB in an Eastern utility company where criterion (job-performance ratings) was so contaminated by tenure that no inferences could be drawn from the study. Other criterion problems indicated in CPAB validation studies have been rater reliability, and restriction of range as a result of preselection of employees. According to Palormo, "The evaluation of computer programmer personnel seems to be particularly difficult" (p. 12).

Jobs in the data processing area are technical and complex. Researchers have tried different ways of arriving at adequate job performance measures. Arvey and Hoyle (1974) attempted to develop behaviorally based rating scales for systems analysts and programmer/analysis approach to determine the essential skills needed by personnel in entry level data processing jobs. Tasks or job analysis is one means of obtaining rater/ratee involvement which, could conceivably result in greater acceptance of the performance rating procedure. The importance of rating employees on objective job-oriented traits has been expressed by several researchers (Buel, 1970; Heier, 1970; Miner, 1968). Sanders and Peay (1974) noted that "rating forms that use ambiguous trait names or descriptions or require ratings on traits which are not observable lead to unreliable results" (p. 33). They recommended, "The list of traits to be evaluated, whether person-oriented, job-oriented, or both, should be determined from a thorough analysis of the jobs to be covered by the evaluation" (p. 29).

In view of previous research into the area of computer programmers selection, a study aimed at finding optimal relationships between predictors and computer programming performance seemed justified. The purpose of this study was to investigate those relationships which exist on various tests of aptitude and programming related abilities. Before a study could be conducted, however, it was decided that a thorough analysis of the job of computer programmer should be conducted. This would serve the dual purpose of developing job performance criteria and guiding the selection of predictors. Tests shown to be valid in past studies would provide an additional, logical basis for the development of an experimental test battery. Once predictors were selected and criteria were developed, their relationships could be investigated by applying the appropriate statistical procedure.

Method

Subjects

Subjects of the study were 38 computer programmers employed by the Data Processing Department of a major food retailing corporation. Individuals held the following job positions: junior programmer, programmer, senior programmer, analyst/programmer, and systems analyst. Programmers who performed systems, teleprocessing, and some software functions were included in the study. The initial sample contained 51 cases. However, 13 individuals were lost form the sample for one of the following reasons.

1. The employee was recently hired and his/her supervisor was not knowledgeable enough of job performance so that a fair rating could be made.

2. An employee who was already rated was not available during the times of test administration. Consequently, the final sample consisted of 38 individuals on whom both test and job performance data were available.

The majority of subjects were Caucasian males. Both minorities and females were represented in the sample, however,

Table 1 displays the composition of the sample by race and sex.

Table 1

Composition of the Validation Sample
By Ethnic Group and Sex

Sex	Ethnic Group		
	Minority[a]	Nonminority	Total
Male	6	25	31
Female	0	7	7
Total	6	32	38

[a]This group consisted of 3 Hispanics, 2 Blacks, and 1 Oriental.

Neither the minority subgroup nor the female subgroup was of sufficient size to allow a separate analysis. Therefore, only the total sample was used in the analysis of test validity. The average of the sample member's age was 32.32 years, with a standard deviation of 6.99. The sample averaged 35.03 months of company service, with a standard deviation of 29.60.

There were no known reasons for believing that the subjects in the present company were not typical of a much larger population of programmers in other business organizations. The company recruits and selects programmers from many different geographical areas within and outside the United States. All contacts with individual subjects in this organization were made through the Personnel Department or the Data Processing Administration Department.

Procedure

Job analysis. A job analysis of the computer programmer position was conducted in order to gain an understanding of the requirements necessary for success in the computer programming job. Results of the job analysis were used to develop job-related performance criteria. In addition, the results of the job analysis served to guide the selection of predictors included in the experimental battery.

The method of job analysis employed in this study was a task checklist approach which focused on the specifications of work tasks and duties and the identification of specific work activities. The task analysis methodology developed by the U.S. Air Force for analyzing, classifying, and describing jobs was the method used in the analysis (Morsh, Madden, & Christal, 1961; Morsh, 1964).

Task inventory statements for data processing jobs were obtained from the Center for Vocational Education, The Ohio State University, Columbus, Ohio. Additional information about the programmer job was provided through organization charts and job descriptions, as well as reports in the psychological and data processing literature (Lyon & Christensen, 1976; Rowan, 1957; Wolfe, 1971).

The preliminary data processing task inventory was reviewed by the company's programmer supervisors (project managers) in several conference sessions. The technical conference mode of gathering job information was chosen since the nature of the programmer job is specialized and complex (Handbook for Analyzing Jobs, 1972). Items were added, modified, or deleted until an inventory of 122 task statements remained. These tasks were then grouped into 12 broad duty categories on the basis of similarity of function among the tasks. The inventory was then administered to 45 incumbent programmers who performed systems and teleprocessing functions. They were asked (a) to indicate which tasks they performed by checking that item, (b) to indicate the relative amount of time spent performing each task, and (c) to add any tasks which they perform but were not already included in the inventory. In a separate conference, project managers were asked to complete the task inventory by specifying the relative difficulty of each task in relation to the total job.

The following criteria formed the basis for the construction of a rating booklet: (a) percentage of incumbents who performed a task; (b) amount of time spent on a task by those performing it; and (c) relative difficulty of the task.

Analysis of the responses to the task inventory revealed no important differences in programming tasks and duties which differentiate systems and teleprocessing programmers. Results indicate that teleprocessing programmers perform a small percentage (1.3%) of tasks in addition to those performed by systems programmers. The majority of programmers perform some of the tasks which comprise each of the 12 categories. It appears that there is no meaningful differentiation among systems and teleprocessing programmers and that no more than one general programmer job exists. Appendix A shows the percentage of programmers performing each of the tasks, the average percent of time spent on each task by those programmers performing the task and the difficulty of each task.

Additional information on the programmer job was generated during scheduled technical conferences and interviewing sessions with programmers and project managers. Results indicate a programmer may be expected to work alone or function as a team member or work on a particular facet of a project. He or she might work under the supervision of another programmer (project leader) or become a project leader. The job demands an attention to detail and accuracy, rather than speed. Time saved by hurriedly designing programs can be lost in the time it takes to debug them repeatedly. While performing accurately, however, a programmer is expected to produce work in sufficient quantities to meet deadlines and maintain an even workflow. Ability to think logically and work in an organized manner was mentioned repeatedly by programmers and their supervisors as a requirement for programming success.

Criteria. The criteria against which the test battery was validated were how well the programmers performed their actual job tasks and duties. Quantifiable measures of job performance were obtained through ratings by each programmer's immediate supervisor. The task inventory served as the basis for the content of the performance rating form. The final rating form contained 109 task statements grouped into 12 duty categories. Seventeen of the 126 original task list statements were eliminated on the basis that

1. They were regarded as very easy to perform (low difficulty rating),

2. They were performed by a small percentage of programmers,

3. Those who performed the task spent a small amount of time on it and it was an insignificant part of their job.

Since so many criteria were to be rated, the risk of rating error appeared particularly great in this study. To offset this effect, a training session was conducted involving all raters in which common rating errors were explained in detail. Specific examples of halo, leniency, and central tendency errors were reviewed and discussed. For each of their programmers, supervisors were provided a booklet containing the programming tasks. The raters were instructed to read through the list of tasks and place a checkmark beside those tasks which they had actually seen a particular programmer perform. Next, supervisors were instructed to rate the performance of that programmer on those tasks he had checked, using a 5-point evaluation scale. Measures of a programmer's performance were calculated by computing the mean rating for each of the 12 duty categories (i.e., summing the ratings for all tasks rated in that duty area). Similarly, an overall (mean) performance rating was obtained for each programmer by averaging all duty scores. When job performance ratings were correlated with test scores, a missing data option was specified in the computer program in order to handle those cases where a programmer was not rated on any of the tasks within a particular duty category.

Predictors. The tests selected for the experimental battery were standardized, commercially published tests. They were selected on the bases of (a) a review of the professional literature concerning their validity in similar studies, (b) job analysis, and (c) informal consultation with incumbents and their supervisors.

The Computer Programmer Aptitude Battery (CPAB), published by Science Research Associates, consists of five subtests designed to measure the following factors:

Subtest	Factor
Verbal Meaning	Communications skill
Reasoning	Ability to translate ideas and operations into mathematical notations
Letter Series	Abstract reasoning
Number Ability	Ability to estimate numerical computations
Diagramming	Ability to analyze problems and order the steps for solution in a logical sequence

A score was obtained for each of the five subtests and for the test as a whole. The CPAB was selected on the basis of favorable reports and reviews it has received in the research literature. The following tests of clerical ability were also included in the battery since they appeared to be measuring factors independent of the factors measured by the CPAB:

1. Short Test of Clerical Ability: STCA-Checking (Science Research Associates).

2. Short Test of Clerical Ability: STCA-Coding (Science Research Associates).

3. Employee Aptitude Survey (EAS) Test Number Four, Visual Speed and Accuracy (Psychological Services, Inc.) Scores on the Test of Supervisory Judgement (Richardson, Bellows, Henry & Co., Inc.) were also available and used in the validation analysis.

All test data were collected on self-scoring answer sheets by the same test administrator. Standardized conditions were insured across subjects. The programmers were tested in groups of three to five and all testing was completed within a period of 2 weeks. Actual testing time took approximately 2½ hours per group. The time between the collection of criterion data and data on the selection measures was approximately 4 to 6 weeks. All data were keypunched on separate cards and all statistical analyses were carried out using standard computer programs both at the company and at North Texas State University.

Results and Discussion

Test performance was reviewed for each test individually by total groups and by subgroups. In Table 2 are provided the predictor means, standard deivations, and *t*-tests for each subgroup present in the sample. In general, there appeared to be no significant difference in scores for either minorities and nonminorities or males and females. Significantly lower scores on the CPAB Letter Series Test were obtained by minority persons when compared to nonminority individuals. On the average, however, females scored significantly higher than males on the same test, which purports to measure abstract reasoning ability. The scores on the CPAB Diagramming Test were significantly lower for minority individuals. Performance on this test is, when contrasted to the Letter Series test, dependent on ability to read and understand written instructions and directions. Results on the Diagramming test are understandable when viewed in light of culture fairness. Results on the Letter Series test are consistent with earlier reports concerning tests of abstract reasoning (Arvey, 1972). Although no tests of fairness was carried out because of small sample size, it is recommended that future research take these subgroup differences, however slight, into consideration.

In Table 3 are shown the intercorrelations among the test scores. Correlations are, for the most part, low to moderate. This indicates that the tests are reasonably dependable and measure somewhat different traits. Each test was, therefore, examined separately in the validation analysis. Tests showing

Table 2

Predictor Means, Standard Deviations, and *t*-values
for Computer Programmer Subgroups

	Minority N = 6		Nonminority N = 32			Female N = 7		Male N = 31			Total N = 38	
	M	*SD*	*M*	*SD*	*t*	*M*	*SD*	*M*	*SD*	*t*	*M*	*SD*
CPAB--Verbal Meaning	15.33	7.31	17.28	7.17	- .61	20.00	8.37	16.29	6.79	1.25	16.97	7.13
CPAB Reasoning	7.33	3.67	10.56	4.14	-1.78	10.14	6.77	10.03	3.54	.06	10.05	4.20
CPAB Letter Series	8.50	2.07	13.81	3.54	-3.54**	16.29	3.15	12.23	3.65	2.72*	12.97	3.87
CPAB--Number Ability	10.00	3.90	11.81	4.39	- .94	9.29	4.82	12.03	4.11	-1.55	11.53	4.32
CPAB Diagramming	23.17	5.19	28.50	5.38	-2.24*	26.00	7.62	28.03	5.18	- .86	27.66	5.63
CPAB--Total	64.33	13.31	81.97	16.06	-2.52*	81.71	24.75	78.61	14.91	.44	79.18	16.81
STCA Checking	32.33	6.80	34.13	6.33	- .63	37.71	4.20	32.97	6.47	1.84	33.84	6.35
STCA--Coding	70.00	7.77	69.50	8.96	.13	72.00	5.42	69.03	9.25	.91	69.58	8.69
EAS--4	105.67	14.81	105.59	16.42	.01	102.29	12.94	106.35	16.68	- .60	105.61	15.98
RBH Supervisory Judgement	147.00	9.14	153.44	8.71	-1.65	152.71	8.36	152.35	9.23	.09	152.42	8.97

*p < .05
**p < .01

Table 3

Intercorrelations Among Tests
N = 38

Test	VM	R	LS	NA	D	T/ CPAB	CH	CO	EAS-4	SJ
Verbal Meaning										
Reasoning	.30									
Letter Series	.11	.38*								
Number Ability	.28	.63**	.13							
Diagramming	.11	.56**	.36*	.36*						
Total CPAB	.63**	.81**	.53**	.68**	.69**					
Checking	.04	.58**	.04	.11	-.08	.04				
Coding	.19	.09	.06	.10	-.33	.07	.42**			
EAS-4	.03	-.10	-.13	.09	-.03	-.03	.46**	.56**		
Supervisory Judgement	.44**	.11	.13	.31	.24	.40*	-.04	.00	-.23	

*p < .05
**p < .01

the highest intercorrelations were tests which seem to measure a general reasoning factor (CPAB Letter Series, Number Ability, and Diagramming). These results are, in general, consistent with reports in the CPAB test manual (Palormo, 1974). Since CPAB total scores are composite of all subscores, a fairly high test/subtest correlation was expected. All subtests were significantly correlated with the CPAB total tests at or greater than the .01 significance level. All three clerical tests, too, correlated significantly with each other. The test of supervisory judgement correlated significantly ($p < .01$) with the Verbal Meaning subtest of the CPAB.

Only one rater was available to rate each programmer's performance. Since no other reliability estimate of the criteria was technically feasible, an internal estimate of reliability was implemented. The reliabilities of the criteria were estimated using Cronbach's Alpha. Results are shown in Table 4. Since the size of the reliability coefficient is affected by the number of items, some duty categories containing only a few items showed low reliabilities. The total rating, however, based on 109 items, was very high indicating similar patterns of responding in different individuals (raters). It is suggested that rater participation in the development of the criteria contributed to the reliable ratings.

Criteria were, like the predictors, statistically analyzed to investigate any differences that could possibly

Table 4

Reliability Estimates (Cronbach's Alpha)
for Criterion Measures

Criterion	Reliability Coefficient
Coding	.36
Designing programs	.85
Designing systems	.86
Testing	.78
Supporting general systems	.74
Maintaining production systems	.90
Changing/Improving production systems	.86
Writing/Maintaining documentation	.83
Scheduling/Directing a project	.88
Advising/Assisting others	.61
Training	.92
Performing clerical tasks	.54
Overall (mean) performance	.96

have been attributed to either race or sex. The means, standard deviations, and _t_-values for each criterion measure are shown in Tables 5 and 6. With the exception of one duty category, Coding, there were no significant differences in ratings between sexes or racial groups.

Quality of performance on specific duties appeared to be closely associated with overall (mean performance when ratings were subjected to a correlational analysis.

Table 5

Criterion Means, Standard Deviations, and *t*-values for Minority/Nonminority

	Minority			Nonminority			
	N	*M*	*SD*	*N*	*M*	*SD*	*t*
Coding	6	2.95	.18	32	3.42	.56	-2.00
Designing programs	6	2.96	.33	31	3.41	.57	-1.86
Designing systems	5	3.32	.32	28	3.36	.57	- .16
Testing	6	3.04	.37	32	3.33	.54	-1.25
Supporting	3	3.00	.00	18	3.07	.82	- .15
Maintaining production systems	5	3.01	.25	27	3.20	.51	- .84
Changing/Improving production systems	5	3.13	.61	27	3.51	.62	-1.28
Writing/Maintaining documentation	5	3.12	.51	26	3.45	.48	-1.38
Scheduling/Directing a project	4	2.66	.76	23	3.23	.55	-1.82
Advising/Assisting others	5	3.33	.71	26	3.36	.46	-.12
Training	1	3.25	.00	12	3.31	.54	
Performing clerical tasks	6	3.25	.48	32	3.35	.45	- .48
Mean performance rating	6	3.05	.25	32	3.34	.34	- .88

Table 6

Criterion Means, Standard Deviations, and *t*-values for Male/Female

	Female			Male			
	N	M	SD	N	M	SD	t
Coding	7	2.97	.51	31	3.43	.53	-2.10*
Designing programs	6	3.19	.31	31	3.36	.59	- .69
Designing systems	5	3.13	.32	28	3.39	.56	-1.01
Testing	7	2.97	.30	31	3.36	.54	-1.84
Supporting general programs	2	3.50	.24	19	3.02	.78	.85
Maintaining production systems	6	3.20	.82	26	3.17	.39	.17
Changing/Improving production systems	6	3.28	.88	26	3.49	.56	- .76
Writing/Maintaining documentation	5	3.55	.72	26	3.37	.45	.71
Scheduling/Directing a project	3	3.03	.16	24	3.16	.64	- .36
Adivsing/Assisting others	6	3.15	.54	25	3.41	.48	-1.15
Training	2	2.88	.18	11	3.39	.53	-1.31
Performing clerical tasks	7	3.36	.30	31	3.33	.48	.20
Mean performance rating	7	3.18	.25	31	3.32	.36	- .41

*$p < .05$

In Table 7 are shown the intercorrelations among all criterion measures. Inspection of this table reveals that except for Duty 8--Writing/Maintaining documentation, Duty 11--Training, and Duty 12--Performing clerical tasks, all criterion measures are substantially intercorrelated. Significant correlations could indicate that if a programmer performs well in one duty category, he performs well in another. Another interpretation is the duty categories overlap to some extent and a certain amount of "halo" error might be evident in the ratings. Low correlations indicate that although a person performs well in one duty category or overall, his or her performance in that duty is independent of other duties.

Supervisory ratings were subjected to an exploratory principal components analysis followed by Varimax rotation. The analysis was undertaken with the intent of finding a relatively small number of linear combinations of factors which would explain the relationships among the 12 variables (Nunnally, 1978). Three factors were extracted from the original 12 variables (duties). The factors represent the performance dimensions underlying the variables. Total variability in the intercorrelation matrix accounted for by these three factors was 71%. In Table 8 are displayed the various rotated factor loadings on the 12 duty categories. Factor I loaded highly on duties 6, 7, 8, and 10 (Maintaining production systems, Changing/Improving production systems, Writing/Maintaining documentation, and Advising/Assisting

Table 7

Intercorrelation Among Criterion Measures
(Sample sizes in parenthses)

	C	DP	DS	T	SGS	MPS	CIPS	WMD	SDP	AAO	TR	PCT
Coding												
Designing programs	.60** (37)											
Designing systems	.42** (33)	.67** (33)										
Testing	.46** (38)	.64** (37)	.69** (33)									
Supporting general systems	.40 (21)	.44* (21)	.17 (20)	.31 (21)								
Maintaining production systems	.52** (32)	.46** (31)	.41* (30)	.54** (32)	.52* (21)							
Changing/Improving production systems	.70** (32)	.68** (31)	.55** (30)	.75** (32)	.34 (21)	.74** (32)						
Writing/Maintaining documentation	.23 (31)	.26 (30)	.09 (29)	.14 (31)	.07 (21)	.03 (31)	.06 (31)					
Scheduling/Directing a project	.26 (27)	.67** (27)	.64** (24)	.52** (27)	-.26 (19)	.40 (24)	.50* (24)	.22 (24)				
Advising/Assisting others	.27 (31)	.46* (30)	.33 (29)	.28 (31)	.29 (20)	.10 (30)	.09 (30)	.60** (29)	.52* (23)			

Table 7--Continued

	C	DP	DS	T	SGS	MPS	CIPS	WMD	SDP	AAO	TR	PCT
Performing clerical tasks	.15 (38)	.14 (37)	.02 (33)	-.04 (38)	-.53* (21)	.00	.19 (32)	.06 (31)	.24 (27)	-.08 (31)	-.01 (13)	
Mean performance	.76**	.86**	.72**	.75**	.46**	.67**	.82**	.36**	.69**	.52**	.49	.21

Table 8

Rotated Factor Matrix (3 Factors Extracted)

Criterion	I	II	III
Coding	.03	-.84	-.15
Designing programs	-.02	-.88	.19
Designing systems	.45	-.73	.13
Testing	.19	-.74	.11
Supporting general systems	.37	-.19	.73
Maintaining production systems	.91	-.16	.18
Changing/Improving production systems	.90	-.26	.14
Writing/Maintaining documentation	.89	.11	.16
Scheduling/Directing a project	.06	-.05	.69
Advising/Assisting others	.78	-.19	.16
Training	.23	-.24	.80
Performing clerical tasks	-.04	-.14	-.67

others). These duties all appear to relate to a Systems Maintenance (Factor I) function. A second factor loaded on variables 1, 2, 3, and 4 (Coding, Designing programs, Designing systems, and Testing). These variables share a common function of what can be called Program Development (Factor II). Extraction of a third factor which loaded on variables 5, 9, 11, and 12 (Supporting general systems, Scheduling/Directing a project, Training, and Performing clerical tasks) can be subsumed under the heading Consulting and Administrating (Factor III). Basically, the three above mentioned factors can be said to explain, in convenient terms, the major job responsibilities of computer programmers. As in most jobs, some overlapping of tasks or duties must be expected. These three factors together comprise the important, independent job functions of a programmer in the present company.

In order to discover relationships between test performance and performance on-the-job, simple (Pearson product-moment) correlations between all predictors and criteria were computed for the total sample. Results are entered in Table 9. The CPAB Reasoning, CPAB Number Ability, and CPAB total show the best single correlations with the criteria. The Verbal Meaning and Letter Series, and Diagramming subtests of the CPAB did not demonstrate a significant relationships with job performance. These results are, generally, with the exception of the Diagramming Test, consistent with earlier findings relating CPAB scores to programming performance. Some possible explanation for the Diagramming result is that

1. All programmers performed well on the Diagramming section and the correlation dropped because of restriction of range on the criterion, and/or

2. Diagramming ability is not an important part of job performance as a programmer.

The latter possibility is supported by programmer comments during the job analysis and administration period. Several individuals reported that they rarely flow-charted (diagrammed) their programs and that more expedient means or shortcuts

Table 9

Test-Criterion Correlations
(Sample sizes in parentheses)

Criterion	CPAB Tests					
	Verbal Meaning	Reasoning	Letter Series	Number Ability	Diagramming	Total
Coding (38)	.19	.49**	.02	.44**	.32*	.43**
Designing programs (37)	.23	.42*	.10	.40*	.00	.32
Designing systems (33)	.11	.26	-.07	.12	.14	.17
Testing (38)	-.01	.34*	.14	.34*	.27	.29
Supporting general systems (21)	-.09	.29	-.01	.26	.02	.11
Maintaining production systems (32)	-.10	.23	.09	.33	.16	.17
Changing/Improving production systems (32)	.01	.49**	.16	.35*	.22	.32
Writing/Maintaining documentation (31)	.00	.00	.21	-.02	-.12	.00
Scheduling/Directing a project (27)	.18	.25	.05	.24	.15	.25
Advising/Assisting others	.21	.17	.00	.29	-.14	.15

Table 9--Continued

Criterion	CPAB Tests					
	Verbal Meaning	Reasoning	Letter Series	Number Ability	Diagramming	Total
Training (13)	.09	-.24	-.33	.05	-.48	-.23
Performing clerical tasks (38)	.12	-.03	.19	-.07	-.01	.06
Mean performance (38)	.13	.43**	.11	.38*	.17	.34*

Criterion	Other Tests			
	STCA Checking	STCA Coding	EAS-4	RBH Supervisory Judgement
Coding (38)	-.12	-.05	-.27	.39*
Designing programs (37)	.06	.39*	.01	.34*
Designing systems (33)	.07	.25	.03	.17
Testing (38)	.07	.28	.10	.29
Supproting general systems (21)	.16	.10	-.05	.02
Maintaining production systems (32)	.10	.17	.07	.23
Changing/Improving production systems (32)	.04	.22	-.04	.18

Table 9--Continued

Criterion	Other Tests			
	STCA Checking	STCA Coding	EAS-4	RBH Supervisory Judgement
Writing/Maintaining documentation	.06	.11	-.15	.00
Scheduling/Directing a project (27)	.15	.41*	.43*	.16
Advising/Assisting others (31)	.14	.30	.17	-.09
Training (13)	.07	.08	.36	-.35
Performing clerical tasks (38)	.09	.26	.09	.17
Mean performance (38)	.12	.35*	.04	.31

$^{*}p < .05$
$^{**}p < .01$

were available to write programs. It is suggested, here, that the CPAB Diagramming test would predict training success better than job performance.

Scores on the STCA Coding test, which involves the ability to memorize symbols and use them accurately were also related with job performance. This result is consistent with job analysis findings that much of the programmer's job involves attention to detail and "clerical" work.

Performance on the RBH Test of Supervisory Judgement is moderately, but not significantly, related to job performance. Examination of the test-test and test-criterion correlation matrices reveals that not only would the above-mentioned tests be useful when used alone, but also they would be more useful if combined, optimally into a battery.

In order to obtain an optimal combination of predictors, a Stepwise Multiple Linear Regression procedure was performed. Factor scores for the three principal components (I-III) and the overall (mean) performance scores were selected as dependent variables. Since the reliabilities of some duty categories were relatively low (Table 4), individual duties were not selected for inclusion as criterion variables in the multiple regression analysis.

All tests were considered equally easy to administer and did not differ markedly in cost. Therefore, the order of variable entry into the predictor models during the various steps was not specified. The CPAB total score was not included in the regression analysis since it is a composite of five other raw subscores in the CPAB battery. Table 10 presents the regression of factor scores and average performance scores on the predictors.

Table 10

Regression of Factors and Overall (Average) Performance on the Predictors[a]

Predictor	Factors I	Factors II	Factors III	Performance
CPAB Reasoning	-.43	.07	.18	.47
STCA Coding	.61	-.28	-.21	.38
RBH Supervisory Judgement	-.10	.40	-.03	.37
CPAB Verbal Meaning	-.02	.09	-.13	-.24
CPAB Letter Series	.17	.23	.19	-.10
CPAB Diagramming	.30	-.12	.14	.06
STCA Checking	-.16	.05	-.26	-.03
CPAB Number Ability	.40	-.39	-.09	.05
EAS Visual Speed and Accuracy	-.29	.35	.41	-.03

[a]The multiple correlations range from .50 to .64 for the total group and criteria. All numbers are standardized regression (beta) weights.

From the regression models, those variables accounting for most variance in the criterion were selected as the best predictors. Due to the sample size of 38, it was decided to limit the retained predictors to a small number to avoid "shrinkage" of the multiple correlation coefficient. The decision to limit the number of variables in the regression equation was based on an F test. The F test indicates whether the Multiple R with more independent variables included is significantly greater than the R with a smaller number of variables (Guilford & Fruchter, 1973). Also, Multiple R's are spuriously high when the ratio of variables to the number of subjects is small and are therefore biased estimators of the variance for the criterion (McNemar, 1969).

In Table 11, multiple correlations between selected tests and the criteria are provided when using overall performance as the criterion, a combination of three tests, CPAB Reasoning, STCA Coding, and RBH Supervisory Judgement, were found to be very highly correlated ($R = .60$). Not only is this relationship statistically significant ($p < .001$) but is also practically significant. All three tests appear to be measuring somewhat independent traits necessary to perform the job of computer programming.

When factor scores were individually regressed on the predictors, somewhat different results were obtained. Two tests (STCA Coding and CPAB Number Ability) were found to be significantly correlated ($p < .05$) with Factor I (Systems Maintenance). The R of .39 was not significantly increased ($F < .05$) by the addition of more predictors. An R of .38 ($p < .05$) between Factor II and two tests (RBH Supervisory Judgement, CPAB Number Ability) was moderately high. Factor III (Consulting and Administrating) did not correlate significantly with any predictor. Out of all predictors, the Diagramming showed the best correlation (.31) with this

Table 11

Multiple Correlations Between Selected Predictors and the Criteria

Criteria	Predictors[a]	Multiple *R*
Overall (mean) Performance	CPAB Reasoning, STCA Coding, RBH Supervisory Judgement	.60**
Factor I (Systems Maintenance)	STCA Coding, CPAB Number Ability	.39*
Factor II (Program Development)	RBH Supervisory Judgement, CPAB Number Ability	.38*
Factor III (Consulting and Administrating)	Diagramming[b]	.31

[a]Addition of more variables did not significantly increase the predictive worth of the regression equation ($F < .05$).

[b]Addition of more variables past one variable did not significantly increase the correlation ($F < .05$).

*$p < .05$
**$p < .001$

factor. It is recommended that the regression equation weights derived in this study be applied to another sample in order to secure a dependable estimate of that predictive worth of the selected variables. McNemar (1969) recommends such a cross-validation process as a check against capitalizing on correlations which might be high because of particular characteristics of the sample at hand.

Results of this study indicate that the job of computer programming requires a number of varying skills and aptitudes.

Results further indicate that commercially available tests, when combined optimally, can be successfully used to help predict a programmer's performance. The Computer Programmer Aptitude Battery appears to be a good measure of the traits necessary for successful performance as a programmer. It is suggested here that good reasoning ability and good ability to work with numbers are both necessary requirements when it comes to performing the job of computer programming. Clerical accuracy is also an important quality, since programmers must pay close attention to detail. A person's ability to cooperate and function as a team member should also be heavily considered by employers when recruiting computer personnel.

As the computer industry changes, so will the requirements. For this reason, it is recommended that further research be carried out using the tests shown to be successful in this study. Considering the present size of the data processing industry, research into the selection of programming personnel is relatively scarce. Not only will the nature of the computer business evolve on an overall basis, but also each company's needs and methods will change. By keeping pace with a dynamic occupational area, researchers in the area of computer programmer selection can offer vital information to all individuals and companies in the data processing industry.

Appendix A

Computer Programmer Tasks: Relative Difficulty, Percentage of Individuals Performing, and Average Percent of Total Time Spent Per Task

Duties/Tasks	Relative Difficulty[a]	Percent of Members Performing	Average Percent of Time
Computer Programmer: Duty A--Coding	3.0	50.0	10.3
1. Consult company programming standards before programming	1.6	68.8	0.7
2. Select standard subroutines and other programming aids	2.3	80.0	0.9
3. Formulate program logic from problem narrative	4.0	77.7	1.7
4. Prepare decision tables, flow charts, diagrams, etc., to depict program logic	3.6	71.1	1.2
5. Use descriptive data names	2.0	82.2	1.5
6. Code programs in ALC	4.0	24.4	0.8
7. Code programs in COBOL	3.0	84.4	3.3
8. Code programs in FORTRAN	b	2.2	0.0
9A. Code programs in PL-1	b	2.2	0.1
9B. Code programs in Mark IV	3.5	2.2	0.0

[a]1 = low difficulty, 5 = high difficulty

[b]Supervisors indicated task was not performed by their programmers.

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
Computer Programmer: Duty B--Designing Programs	3.3	56.2	12.3
10. Analyze proposed computer applications	4.0	60.0	1.3
11. Analyze documents, files, and reports related to workflow	2.8	64.4	1.2
12. Conduct primary research to determine compatability of proposed computer equipment with present applications/equipment	3.0	20.0	0.3
13. Consult with project programmers to define program objectives	3.4	73.3	1.3
14. Consult with project programmers to determine interface with existing programs and systems	3.2	77.7	1.3
15. Design program specifications for individual programs within the system	3.6	68.8	1.9
16. Design record formats, data access methods, print lines, etc.	2.8	93.3	2.1
17. Formulate standards for programming	3.0	26.6	0.4
18. Determine programs to be written to obtain system results	3.6	66.6	1.3
19A. Organize programs for system implementation	3.6	64.4	1.2
19B. Design effective "quickie" programs for own use	2.8	2.2	0.0

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
Computer Programmer: Duty C Designing Systems	3.3	49.6	10.6
20. Determine file organization and device type	3.4	75.5	1.2
21. Gather data for analysis of problem areas	3.0	60.0	1.3
22. Develop system flow chart	3.4	55.5	1.0
23. Define computer controls and input/output specifications	3.8	55.5	1.0
24. Make cost analysis	3.3	15.5	0.2
25. Compare proposed system requirements with machine capabilities	3.8	24.4	0.4
26. Check existing procedures for efficiency	2.6	42.2	0.6
27. Develop back-up and recovery capabilities for system	3.4	66.6	1.1
28. Evaluate information submitted by user for systems specification	3.8	60.0	1.4
29. Meet with user staff to design and implement automated systems	3.6	55.5	1.3
30. Meet with operations department to determine interface with their schedule	2.5	33.3	0.4
31. Review input/output forms with user and interdepartmental personnel	2.6	51.1	0.7

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
Computer Programmer: Duty D--Testing	3.3	77.8	17.5
32. Assess system for efficiency	3.6	57.7	0.6
33. Assemble test data to check-out program	3.0	93.3	2.0
34. Prepare JCL deck, JCL procs for testing	2.6	95.5	1.9
35. Test program for logic correctness	3.4	95.5	2.8
36. Analyze program performance during testing/execution	3.4	80.0	1.6
37. Review testing results for logic correctness	3.0	97.7	2.3
38. Debug program	4.0	97.7	2.5
39. Compare actual system output with desired system output	3.0	86.6	1.6
40. Make timing studies	3.0	42.2	0.5
41. Determine response time of a system in a peak load situation	4.5	17.7	0.3
42. Implement programs into production	2.8	91.1	1.4
Computer Programmer: Duty E--Supporting General Systems	3.5	14.5	3.4
43. Make inquiries on computer terminal	1.7	51.1	1.3
44. Conduct benchmark activities	4.5	11.1	0.2

Duties/Tasks	Relative Difficulty[a]	Percent of Members Performing	Average Percent of Time
45. Develop procedures for loading/changing/calling up program on computer	2.0	13.3	.03
46. Install new software releases	5.0	6.7	0.3
47. Review literature for new techniques	2.3	11.1	0.1
48. Maintain teleprocessing software	5.0	13.3	0.3
49. Identify alternatives to the current work process for increased efficiency	3.8	26.6	0.4
50. Prepare technical comparison of alternative terminal systems	4.0	4.4	0.1
51. Develop input keying statistics for a terminal system	2.3	11.1	0.2
52A. Formulate procedures for maximum utilization of teleprocessing network	4.5	8.9	0.2
52B. Maintain software release	b	2.2	0.1
Computer Programmer: Duty F--Maintaining Production Systems	2.5	57.4	9.1
53. Read documentation to become familiar with programs	2.0	80.0	1.3
54. Maintain control of data sets	2.6	57.7	0.7
55. Maintain systems for maximum utilization	3.4	31.1	0.6

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
56. Read error printout	2.0	62.2	0.8
57. Determine if problem is hardware or program error	3.6	77.7	1.1
58. Restart job with restart procedures	2.2	48.8	0.5
59. Initiate printing of special/periodic computer runs	2.2	44.4	0.6
60. Review system output for maximum computer utilization	3.6	28.8	0.4
61. Contact user for assistance concerning systems problems	2.2	68.8	1.0
62. Notify correct programmer in case of program malfunction	1.2	42.2	0.4
63. Write troubl reports on computer problems and action taken	1.6	73.3	0.7
64. Research documentation to solve problems that arise within the system	3.0	73.3	1.1
Computer Programmer: Duty G--Changing/Improving Production Systems	3.2	70.7	7.8
65. Analyze/review requests for program change	3.0	62.2	1.1
66. Make design changes on existing systems	3.6	66.6	1.3
67. Correct JCL programming problems	2.2	77.7	0.8
68. Solve program problems	3.6	91.1	2.1

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
69. Make program changes for correction of error conditions	3.6	91.1	2.0
70. Recommend revisions to standard operating procedures	3.0	46.6	0.5
Computer Programmer: Duty H--Writing/Maintaining Documentation	2.5	50.2	8.1
71. Log file changes	1.4	26.6	0.3
72. Log program changes	1.4	40.0	0.5
73. Write program narrative and maintain program documentation	3.0	71.1	1.2
74. Write and maintain systems documentation	3.2	68.8	1.2
75. Write and maintain operations documentation	2.2	71.1	1.3
76. Write and maintain user guides	2.6	53.3	1.0
77. Review documentation for completeness/accuracy	2.0	66.6	0.9
78. Develop standards and procedures for documentation	3.0	15.5	0.2
79. Update programs to comply with the law, O.S. system changes, and user requests	2.8	66.6	1.3
80. Compile/maintain standard operating procedures manual	3.0	22.2	0.2

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
Computer Programmer: Duty I Scheduling/Directing a Project	3.2	26.2	3.8
81. Determine priority of tasks	3.0	35.5	0.4
82. Set target dates for work completion	3.5	35.5	0.5
83. Schedule new projects	3.3	13.3	0.2
84. Determine area of responsibility for other programmers	3.7	26.6	0.4
85. Assign and schedule work of other programmers	3.8	24.4	0.3
86. Assess programming effort for new systems	3.5	44.4	0.9
87. Recommend changes in work processes/procedures	3.3	37.7	0.5
88. Assist in scheduling system projects with the user department	3.3	24.4	0.4
89A. Schedule programming projects with other sections of the department	3.3	17.7	0.3
89B. Attend project meetings	1.0	2.2	0.0
Computer Programmer: Duty J--Advising/Assisting Others	3.1	27.7	3.6
90. Make recommendations regarding job applicants	2.0	11.1	0.1
91. Make notes/comments in program to help other personnel understand program logic	2.0	53.3	0.8

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
92. Give technical assistance to programming personnel	3.4	53.3	0.8
93. Give technical assistance to application project personnel for any questions associated with an on-line system	4.0	15.5	0.2
94. "Sell" system plan to user	4.0	20.0	0.3
95. Assist in providing cost/capability/benefit information on system to prospective users	3.5	8.9	0.1
96. Advise user personnel regarding remote hardware utilization and troubleshooting malfunctions	3.5	11.1	0.1
97. Advise user personnel regarding their use of automated systems including both on-line and batch systems	3.4	26.6	0.5
98. Prepare letters to user for project manager	2.5	48.8	0.7
Computer Programmer: Duty K--Training	2.8	15.6	1.5
99. Identify need for training programmers/operations/users	2.8	26.6	0.3
100. Plan on-the-job training for programmers/operations/users	3.5	15.5	0.2
101. Develop on-the-job training for programmers/operations/users	3.5	11.1	0.2

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
102. Conduct on-the-job training for programmers/operations/users	3.0	22.2	0.4
103. Evaluate training effectiveness	2.8	8.9	0.1
104. Attend training schools to receive programming instruction	1.5	8.9	0.2
Computer Programmer: Duty L--Performing Clerical Tasks	1.2	55.1	11.5
105. File correspondence and interoffice memos	1.0	46.6	0.4
106. File operations documentation	1.0	51.1	0.5
107. File user guides	1.0	44.4	0.5
108. File JCL cards	1.0	42.2	0.4
109. File/update technical references manuals	1.0	48.8	0.4
110. File program listing	1.0	86.6	0.8
111. Distribute letters, documentation, etc., to necessary personnel	1.0	55.5	0.5
112. Prepare/distribute monthly weekly status reports	1.0	71.1	0.7
113. Pickup/distribute output within the department and other departments	1.0	37.7	0.4
114. Complete and submit time reports	1.4	77.7	0.7

Duties/Tasks	Relative Difficulty	Percent of Members Performing	Average Percent of Time
115. Deliver input ot operations or pickup desk	1.0	44.4	0.4
116. Enter data/commands on terminal	2.0	64.4	2.1
117. Remove/replace print ribbon, paper, etc., on terminals	1.0	6.7	0.1
118. Proofread typing	1.2	48.8	0.5
119. Answer telephone	1.0	93.3	1.3
120. Photocopy forms, letters, etc.	1.0	64.4	0.6
121. Keypunch	1.4	77.7	0.7
122. Bind printouts	1.0	46.6	0.4

References

Alspaugh, C. A. Identification of some components of computer programming aptitude. Journal for Research in Mathematics Education, 1972, 3, 89-98.

Alspaugh, C. A. A study of the relationships between student characteristics and proficiency in symbolic and algebraic computer programming. Dissertation Abstracts International, 1970, 31:4627B.

Arvey, R. D. Some comments on culture-fair tests. Personnel Psychology, 1972, 25, 433-448.

Arvey, R. D., & Hoyle, J. C. A Guttman approach to the development of behaviorally based rating scales for systems analysts and programmer/analysts. Journal of Applied Psychology, 1974, 59, 61-68.

Bauer, R., Mehrens, W. A., & Vinsonhaler, J. F. Predicting performance in a computer programming course. Educational and Psychological Measurements, 1968, 28, 1159-1164.

Biamonte, A. J. A study of the effect of attitudes on the learning of computer programming. In Malcolm H. Gotterer (Ed.) Proceedings of the Third Annual Computer Personnel Perconnel Research Conference. Silver Spring, Maryland: Computer Personnel Research Group, 1965.

Buel, W. Items, scales, and raters: Some suggestions and comments. Personnel Administration, 1970, 33(3), 26-30.

Buros, O. K. (Ed.) The seventh mental measurements yearbook. Highland Park, N. J.: Gryphon, 1972.

Correnti, R. J. Predictors of success in the study of computer programming at two-year institutions of higher education. Dissertation Abstracts International, 1969, 30, 3718A.

Cronbach, L. J. Essentials of psychological testing (third edition). New York: Harper and Row, Publishers, Inc., 1970, 417-421.

Denelsky, G. Y., & McKee, M. G. Prediction of computer programmer training and job performance using the AABP test. Personnel Psychology, 1974, 27, 129-137.

Guilford, J. P., & Fruchter, B. Fundamental statistics in psychology and education (fifth edition). New York: McGraw-Hill, 1973.

Guion, R. M. Personnel testing. New York: McGraw-Hill, 1965.

Gordon, B. F., & Dennis, R. A. Characteristics and performance predictors of 7094 computer service operators. Proceedings of the Annual Computer Personnel Research Conference, 1966, 4, 96-106.

Handbook for analyzing jobs. U. S. Department of Labor (Manpower Administration), Washington, D. C., 1972, Stock Number 2900-0131.

Heier, W. Implementing an appraisal by results program. Personnel, 1970, 47, 24-32.

Hollenbeck, G. P., & McNamara, W. J. CUCPAT and programming aptitude. Personnel Psychology, 1965, 18, 101-106.

Howell, M. A., Vincent, J. W., & Gay, R. A. Testing aptitude for computer programming. *Psychological Reports*, 1967, 20, 1251-1256.

IBM. *Manual for administrating and scoring the Aptitude Test for Programmer Personnel*. White Plains, New York: IBM Technical Publications Department, 1964.

John, E. R., & Miller, J. G. The acquisition and application of information in the problem-solving process. *Behavioral Science*, 1957, *2*, 291-300.

Katz, A. Prediction of success in automatic data processing course. U. S. Army Personnel Research Office Technical Research Note, 1962, NO. 126.

Lyon, R. L., & Christensen, J. E. A task analysis approach to developing data processing curricula. *Journal of Educational Data Processing*, 1976, *13*, 2-12.

McNamara, W. J., & Hughes, J. L. A review of the selection of computer programmers. *Personnel Psychology*, 1961, *14*, 39-51.

McNemar, Q. *Psychological Statistics*. New York: John Wiley, 1969.

Martin, M. A. A study of the concurrent validity of the Computer Programmer Aptitude Battery. *Studies in Personnel Psychology*, 1971, *3*, No. 1, 69-76.

Miller, R. L. A study of the relationship which exists betwen achievement in high school vocational data processing programs and the Programmer Aptitude Test.

Masters Thesis, North Carolina State University (Raleigh, N. C.), 1970.

Miner, J. Management by appraisal: A capsule review and current references. Business Horizons, 1968, 11, 83-94.

Morsh, J. E. Job analysis in the United States Air Force. Personnel Psychology, 1964, 17, 7-17.

Morsh, J. E., Madden, J. M., & Christal, R. E. Job analysis in the United States Air Force. WADD-TR-61-113, AD-259389 Lackland AFB, Texas: Personnel Research Laboratory, Wright Air Development Division, February, 1961.

Nunally, J. C. Psychometric theory (second edition). New York: McGraw-Hill, 1978.

Oliver, T. C., & Willis, W. K. A study of the validity of the Programmer Aptitude Test. Educational and Psychological Measurement, 1963, 23, 823-825.

Palormo, J. M. Computer Programmer Aptitude Battery. Science Research Associates, Inc., Chicago, Illinois, 1974.

Perry, D. K. Training and job perfomrance: Validities of programmer trainee selection variables. Santa Monica, California: System Development Corporation, 1964 (Tech. Memo. 2172).

Perry, D. K., & Cannon, W. M. Vocational interests of computer programmers. Journal of Applied Psychology, 1967, 51, 28-34.

Rowan, T. C. Psychological tests and selection of computer programmers. Journal of the Association for Computing Machinery, 1957, 4, 348-353.

Sanders, M. S., & Peay, J. M. Employee performance evaluation and review: A summary of the literature. Naval Ammunition Depot Technical Reprot, 1974 (Aug.), RDTR No. 282.

Upshall, C. C., & Riland, L. H. An unpublished study at Eastman Kodak Company, 1958. Cited by W. J. McNamara and J. L. Hguehs, a review of research on the selection of computer programmers. Personnel Psychology, 1961, 14, 39-51.

Wolfe, J. M. A new look at programming aptitudes. Business Automation, 1970, 17, 36-45.

Wolfe, J. M. Perspectives on testing for programming aptitude. Association for computing machinery, Proceedings, 1971 Annual Conference, Chicago, Illinois, 268-327.