PREDICTION OF SUCCESS IN COMPUTER PROGRAMMING COURSES

APPROVED:

Earl W Kooker
Major Professor

Harold D. Holloway
Minor Professor

Harold D. Holloway
Director of the Department of Psychology

Robert B. Toulouse
Dean of the Graduate School

Dempsey, Betty J., *Prediction of Success in Computer Programming courses*. Master of Arts (Clinical Psychology), May, 1971, 43 pp., 8 tables, bibliography, 25 titles.

This study dealt with the problem of investigating the validity of two recently developed paper-and-pencil tests of programmer aptitude, the *Aptitude Test for Programmer Personnel* (ATPP) and the *Computer Programmer Aptitude Battery* (CPAB), in predicting success in computer programming. The ATPP and CPAB were administered to 57 programming students enrolled in COBOL and FORTRAN classes at a junior college in a large southwestern city. The criterion of performance in the programming classes consisted of the final class grade based on numerical scores on class tests and programming performance. COBOL and FORTRAN grades were placed on a comparable scale by converting the final numerical grades into stanines. The four ATPP and six CPAB scores served as predictor measures.

The study involved correlating the scores of each of the ATPP and CPAB predictor measures with the stanine transformation of grades using the Pearson product-moment correlation. A step-wise multiple regression analysis was also performed to determine whether the ATPP and CPAB Total scores taken together would significantly increase the validity coefficient.

Analysis of the results presented for all 57 students revealed that the ATPP Total scores, as well as the three ATPP subtests, were significantly correlated with grades at the .01 level. The correlation between the CPAB Total scores and grades was significant at the .01 level. Only two of the CPAB subtest correlations failed to reach significance at the .01 level, but both of these were significant at the .05 level.

These results supported the two hypotheses in the study. The first hypothesis that a significant relationship would exist between the ATPP Total scores and grades was accepted at the .01 level, as was the second hypothesis that there would be a significant relationship between the CPAB Total scores and grades.

The results of the step-wise multiple regression analysis yielded a low and nonsignificant F value (p>.05). Thus, the combination of the ATPP and CPAB Total scores did not significantly improve prediction of the criterion.

Data for the FORTRAN and COBOL students were also analyzed separately. This analysis consisted of computing separate Pearson product-moment correlations between the criterion and the predictor measures for the 22 FORTRAN and 35 COBOL students.

All ATPP correlations for the COBOL students were significant at the .01 level; and with the exception of one ATPP subtest, all of the ATPP correlations for the FORTRAN students were significant at the .05 level. Five of the CPAB correlations for the COBOL students were significant at the .01 level, and one was significant at the .05 level. However, none of the CPAB scores were significantly correlated with FORTRAN grades.

Comparison of the data from the separate analysis of the COBOL and FORTRAN students with the data obtained from the entire sample of 57 students revealed that the ATPP consistently yielded higher correlations with programming grades than the CPAB. Furthermore, ATPP Total scores provided the most consistent single predictor of programming success in the study.

It was concluded that the evidence tends to support the applicability of the ATPP across different programming situations, and further supported the practicality of efforts to refine such paper-and-pencil tests of programming aptitude. However, the failure to obtain significant correlations between the CPAB and FORTRAN grades raised the question of the effectiveness of this measure in various types of programming. Cross-validation was, therefore, recommended to determine whether the applicability of the CPAB in predicting relative performance is limited only to certain kinds of programming.

PREDICTION OF SUCCESS IN COMPUTER PROGRAMMING COURSES

THESIS

Presented to the Graduate Council of the

North Texas State University in Partial

Fulfillment of the Requirements

For the Degree of

MASTER OF ARTS

By

Betty J. Dempsey, B. S.

Denton, Texas

May, 1971

LIST OF TABLES

Table | Page

1. Means and Standard Deviations for Numerical Grades and Equivalent Stanines. 25

2. Means, Standard Deviations, and Correlations of the Aptitude Test for Programmer Personnel with the Stanine Transformation of Final Course Grades for All 57 Programming Students. 26

3. Means, Standard Deviations, and Correlations of the Computer Programmer Aptitude Battery with the Stanine Transformation of Final Course Grades for All 57 Programming Students. 27

4. Results of the Step-Wise Multiple Regression Computed between Grade Stanines and Total Scores for the ATPP and CPAB. 28

5. Means, Standard Deviations, and Correlations of the Aptitude Test for Programmer Personnel with the Stanine Transformation of COBOL Grades 30

6. Means, Standard Deviations, and Correlations of the Aptitude Test for Programmer Personnel with the Stanine Transformation of FORTRAN Grades 31

7. Means, Standard Deviations, and Correlations of the Computer Programmer Aptitude Battery with the Stanine Transformation of COBOL Grades 33

8. Means, Standard Deviations, and Correlations of the Computer Programmer Aptitude Battery with the Stanine Transformation of FORTRAN Grades 34

PREDICTION OF SUCCESS IN COMPUTER PROGRAMMING COURSES

The prediction of academic performance at the college level has been the focus of interest for a large number of studies. Typically, investigators in this area have been concerned with the prediction of over-all grade point averages or with final course grades in a particular field of endeavor. In a recent typical study using grade point average as the performance criterion, Distefano and Rice (1966) investigated the predictive validity of the School and College Ability Test (SCAT). The SCAT was administered to 698 college students at Louisiana College. First-year grade point averages were obtained for all 698 students, but four-year grade point averages were available for only 110 of the students. They found correlations of .16 (p>.05), .48 (p<.01), and .48 (p<.01) between first-year grade point averages and scores on SCAT Quantitative, Verbal, and Total, respectively. Correlations between four-year grade point averages for the 110 students in the sample and the SCAT Quantitative (r=.38), Verbal (r=.68), and Total (r=.61) were all statistically significant. Boyce (1964) was interested in the prediction of final course grades in a college algebra course. He found that the Mathematics Usage Test and the Cooperative Mathematics Pre-Test for College Students correlated .25 ($p<.05$) and .41 ($p<.01$), respectively, with the final course grades of 100 students enrolled in a college algebra course. Recent reviews (Lavin, 1966) on the relationship between aptitude variables and performance criteria have indicated that the average correlation between college level aptitude tests and grade point averages is about .50.

Although a vast amount of experimentation has been done concerning prediction of performance in the traditional academic courses, such as mathematics or English, comparatively few studies have been conducted in the area of computer programming courses. This may be due, at least in part, to the relative recency of the addition of programming courses to university and college curriculum. Valid predictors of performance, however, are just as badly needed in the field of programming as in any other field.

The need for valid predictors of programming performance is most apparent in the industrial setting. In industry, a wide variety of different tests are employed in screening applicants for programmer positions. The majority of these tests were designed for purposes other than programmer selection, and they are rarely validated against programming performance criteria. In a survey of programmer selection procedures, Watson (1961) sent survey questionnaires to 262 business firms in the United States and Canada.

Of the 262 firms surveyed, about 52%, or 136 firms, replied. Watson found that 82% of the respondents, or 96 of the 136 firms, reported that they used psychological tests in programmer selection. Replies to the survey questionnaire indicated that 29 different tests were used. While only 2 of these tests had been developed specifically for testing programmer aptitude, 20 were conventional psychological or personnel tests designed for purposes other than differentiating between persons with high and low potential for programming work. In a similar survey of 500 organizations, the System Development Corporation (SDC) conducted a National Survey of Digital Computing Personnel in 1962 (Perry, 1962). The 1962 survey obtained a 50% return, i.e., 250 organizations responded, and of these, about 57%, or 137 of the organizations, reported the use of tests in the selection of programmers and programmer trainees. These 137 organizations reported using over 60 different tests in programmer selection. Perry found, however, that only 3 of these organizations had conducted validation studies, and that in each of these cases a specially designed programmer aptitude test had been validated rather than an existing psychological or personnel test.

Among the relatively few conventional psychological tests which have been validated against programming performance, the Thurstone Primary Mental Abilities Test (PMA) has probably been the most extensively studied. The predictive validity of the PMA has been investigated by the Rand Corporation (Rowan, 1957) and by SDC (Perry, 1964). Rowan (1957) reported a multiple correlation of .54 between supervisory ratings of programmer performance and scores on the Verbal Meaning and Reasoning subtests of the PMA and the Emotional Stability subtest of the Thurstone Temperament Schedule.

Perry (1964) investigated the relationship between scores on the PMA and Otis Test of Mental Ability, Higher Examination (OTMA), training grades, and subsequent job performance criteria. He conducted a longitudinal study of 407 male and 45 female programmer trainees from 44 different training classes. Scores on the OTMA and PMA Verbal Meaning, Space, and Reasoning subtests were used as predictor variables. The criterion variables included training class grades and 4 indices of job success: tenure, salary progress, salary review ratings, and job classification progress. Perry found that PMA and OTMA scores were significantly, and approximately equally, related to training course grades, but the OTMA correlations did tend to vary less from class to class than the PMA. When the predictor variables were correlated with the 4 job criteria, low and inconsistent validity coefficients resulted. Significant correlations were found, however, between training grades and job criteria. In a review of computer programmer selection and training procedures at SDC, Perry and Cantley (1965) described several other research projects by SDC which dealt with the relationship between the PMA and programming performance of trainees. They reported that, typically, the combined scores of the PMA Reasoning and Verbal Meaning subtests have correlated about .50 with programmer training criteria.

In a study similar to that conducted by Perry (1964), Sweetland (1962) did a longitudinal follow-up of 755 programming trainees in 9 different 8-week classes. He analyzed the relationship between 2 predictor variables, class grades, and subsequent job performance criteria. The predictor variables were scores on the OTMA and estimates of the trainees' motivation, defined as drive to learn programming. Criterion variables included final grades and on-the-job ratings by supervisors. The trainees were given the OTMA and were ranked according to their motivation by their class instructors and by their classmates. Thus, 2 estimates of motivation were obtained: an instructor and a peer estimate. Using Kendall's Tau as the measure of correlation, Sweetland found that the OTMA scores tended to be significantly related to class grades with correlations ranging from .23 to .60. Significant correlations ranging from .34 to .82 were found between instructor estimates of motivation and class grades. Correlations ranging from .57 to .82 between the peer estimates of motivation and class grades also turned out to be significant. The correlations were higher and the variability lower among the peer estimates than among the instructor estimates, and correlations between the peer estimates and class grades were higher than those between OTMA scores and class grades. However, tests of significant differences between correlations were not performed. Sweetland also found that the OTMA scores were much less valid for predicting job performance than training performance, but there were positive relations between training grades and job performance criteria. Thus, both Perry (1964) and Sweetland (1962) found significant correlations between conventional psychological tests and training grades and between training grades and job performance criteria, but neither investigator found these tests to be consistently valid predictors of job performance. Yet, according to the surveys (Perry, 1962; Watson, 1961) cited above, a wide variety of psychological tests which have never been validated against programming performance criteria are being used to screen applicants for programmer jobs.

In the exploratory stages of the development of tests designed especially to measure aptitude for programming, experimenters dealt with the problems of (a) whether or not programmers actually differed significantly from non-programmers, and (b) the extent and nature of such differences. Roemmich (1963) performed an experiment to determine whether computer programmers could be differentiated from non-programmers by means of several aptitude tests. Thirty non-programmers, persons with jobs other than that of programmer, and thirty-four programmers were administered the Employee Aptitude Survey (EAS) and the Flanagan Aptitude Classification Tests (FACT). The performance of the programmers and the non-programmers differed significantly on 4 of the subtests--Numerical Ability (t=3.40, p<.01), Symbolic Reasoning (t=3.31, p<.01), and Numerical Reasoning (t=3.25, p<.01) of the EAS, and Tables (t=2.50, p<.01) of the FACT.

Biamonte (1965) wanted to know whether attitudes were related to performance in a computer programming course. He administered 3 attitude measures--the Rokeach's Dogatism Scale, the McClosky's Conservatism Scale, and the California F-Scale of Authoritarianism--to 201 computer programmer students. Biamonte found that the attitude scores were negatively but non-significantly related to course grades.

Perry and Cannon (1965, 1967, 1968) explored the area of programmer interest to see if they differed from those of people in other professions. They developed 2 programmer interest keys for the Strong Vocational Interest Blank. One of the keys was developed to measure the interests of male programmers, whereas the other was designed for female programmers. Perry and Cannon administered the keys to 1,378 male and 293 female programmers. They found that, compared to men and women in other professions, computer programmers were less interested in people and more interested in mathematics and problem solving. They also found that scores on the programmer interests keys were positively related to satisfaction with the occupation. Perry (1967) observed, however, that while programmers dissatisfied with the programming profession scored significantly lower on the keys than satisfied programmers, satisfaction was not related to salary progress in the programming field.

In an effort to identify the performance characteristics of successful programmers, a few experimenters have made exploratory surveys of the opinions of programmers and programmer supervisors. Peres and Arnold (1963) asked 23 programmers and 6 programmer supervisors to describe, by way of a written essay, the behavior of the best programmer they had ever known. When a content analysis was performed on the essays, 140 statements describing programmer behavior emerged. These statements were sorted into 12 homogeneous groups and arranged in the form of a descriptive check list. Each programmer and supervisor was given two programmer description check lists upon which he was asked to rate, on a 5-point scale, the best and worst programmer he had ever known. By means of the Wherry-Winer method of factoring large numbers of items, the data obtained from the 58 check lists were subjected to a modified centroid factor analysis. From this analysis, Peres and Arnold found that the following 6 characteristics emerged as important in programming performance:

1. Personal maturity and stability. The good programmer should have a stable and mature personality, as exhibited by the ability to work under pressure, to assume the responsibility for his own behavior, and to take suggestions and criticism in their proper perspective.

2. Cooperation in interpersonal relations. The successful programmer is a good listener. He has the ability to work with, and to gain the cooperation of, other people.

3. Communication skills. The good programmer has the ability to speak and write clearly, and to understand and simplify technical terms.

4. Thoroughness and dependability. This factor characterizes the programmer who becomes thoroughly acquainted with the problem and attends to every detail before he begins to program. That is, he has the ability to pay attention to relevant details while simultaneously maintaining a

breadth of perspective regarding the problem as a whole.

5. Job interest and zeal. The successful programmer is one who is enthusiastic and innovative in his work. He has the ability to both accept and develop original and creative ways of programming.

6. Professional competence. The good programmer has the ability to analyze a problem from beginning to end, to reason and think logically, and to maintain technical proficiency.

In a review of the research on the selection of computer programmers, McNamara and Hughes (1961) have found considerable agreement among investigators that, out of all the various characteristics required of successful programmers, the ability to reason constitutes the most important single characteristic. Two different types of tests, both of which presumably measure reasoning ability, have been developed specifically for testing programmer aptitude: an apparatus type test, and a paper-and-pencil type test. As might be expected, programmer aptitude tests, especially the paper-and-pencil type, have a more complete history of validation with programming performance than either the conventional psychological or the general personnel tests.

Regarding the developmental history of the apparatus type test of programmer aptitude, John and Miller (1957) reported their development of an apparatus called Problem Solving using Information (PSI) whereby human reasoning and problem solving could be objectively and quantitatively assessed. The PSI apparatus displays a panel of stimulus lights to the S. The S's task is to activate a particular light on the panel. In order to accomplish this task, the S must experiment with the interrelationships between the stimulus lights and activate the lights in a particular sequence and combination.

Langmuir (1960) designed an apparatus similar to the PSI which he called the Logical Analysis Device (LAD). The LAD is similar to the PSI in that both are apparatus tests designed to measure reasoning and problem solving ability by requiring the S to solve problems using stimulus lights. Nine of the lights form a circular display around one center light. The S's task is to activate the center light. In order to solve the problem, the S must first discover the rules for responding logically in a correct sequence. In a follow-up study of several groups of UNIVAC and IBM programmers, Langmuir (1960) reported correlations ranging from .50 to .88 between the LAD and programming job performance criteria.

Opler (1963) reported the development of a computer-based apparatus test called the Computer Usage Company Programmer Aptitude Test (CUCPAT). The administration of the CUCPAT requires a computer and a CUCPAT computer program, but the problems to be solved are essentially the same as those used in the LAD (Langmuir, 1960). The CUCPAT requires the S to solve logical problems by activating sequences of lights on a 1401 IBM computer console. In a validation study of the CUCPAT, Opler (1963) administered the CUCPAT to a sample of 20 subjects. This sample consisted of 3 managers, 2 systems analysts, 5 programmers, 1 typist, 1 computer operator, 1 technical writer, 4 secretaries, and 1 clerk. The subjects were then ranked according to their programming aptitude by 6 supervisors, and their CUCPAT scores were placed in rank-order. When a rank-difference correlation was computed between the rank-order of the CUCPAT scores and supervisor's mean rank of the 20 subjects, Opler found a coefficient of .48 ($p < .05$). Hollenbeck and McNamara (1965) observed, however, that the relationship between the CUCPAT and supervisor's rank disappears when the non-programming personnel are eliminated and only the programmers and systems analysts are considered.

In an effort to find a predictor of programming performance, several paper-and-pencil type tests of programmer aptitude have been developed. Among such tests, the Programmer Aptitude Test (PAT), developed at IBM by McNamara and Hughes (1961), has been the most widely used. The test contains 3 parts: a number series, a figure analogy series, and an arithmetic reasoning series. The PAT items are simliar to those in the Q-part of the American Council on Education Psychological Examination for College Freshmen (ACE).

In a validation study of the PAT, McNamara and Hughes (1961) found a product-moment correlation of .50 between the PAT scores and the final grades of 245 trainees in IBM 650, 702, and 705 programming classes. When McNamara and Hughes (1961) investigated the relationship between scores on the PAT and the technical job performance of 52 IBM 702 and 705 programmers, they found a correlation of .36 (p<.05) between the PAT and manager's ratings on a 5-point scale of technical performance.

McNamara and Hughes (1961) reported that Rush found the PAT scores of 161 programmers at the Standard Oil Company were significantly related to supervisor's ratings of learning ability (r=.33), of technical skills (r=.28), of imagination and ingenuity (r=.37), and of over-all performance (r=.31). They (McNamara & Hughes, 1961) also reported that Upshall and Riland found that supervisor's ratings of the job performance of 13 programmers at Eastman Kodak Company were significantly related (p<.05) to the PAT (Spearman rho of .61) and to the Brown-Carlsen Listening Comprehension Test (Spearman rho of .60).

In an attempt to determine a predictor of success in an Army programming course, Katz (1962) gave the PAT plus several tests from the Army Classification Battery (ACB) to 190 programmer trainees. When the test scores were correlated with final course grades, he found that a combination of arithmetic and verbal reasoning tests from the ACB correlated .61 with final grades. The PAT yielded a correlation of .67 with final grades. A multiple correlation of .68 was found between these tests and final course grades. Thus, the PAT correlated higher (r=.67) with final course grades than any other single measure in the study.

Howell et al. (1967) administered the PAT, the Color Naming Test, the General Aptitude Battery, and the Federal Service Entrance Examination (FSEE) to 2 samples (N=135, and N=118) of Civil Service employees at the United States Public Health Service. They found that Part II, the figure analogy series, and Part III, the arithmetic series, of the PAT to be the best predictors of a simulated work sample of programming. They also found that in one of the samples (N=118), the addition of the Numerical part of the FSEE to Part I and II of the PAT significantly improved prediction of the criterion (R=.71, N=118; R=.60, N=135).

In 1959, a revised form of the PAT, the Revised Programmer Aptitude Test (RPAT), was developed at IBM by

McNamara and Hughes (1961). Although the RPAT was found somewhat more difficult than the PAT, the 2 forms are quite similar ($\underline{r}$=.88). When the RPAT was validated against the final class grades of 175 programming students in IBM 650 and 7070 programming classes, McNamara and Hughes (1961) found product-moment correlations ranging from .34 to .85 for 11 classes. The correlation was .51 for all classes combined. These same investigators (McNamara & Hughes, 1961) found the RPAT to be significantly related ($\underline{r}$=.44, $\underline{p}$<.05) to supervisor's rankings of the over-all job performance of 41 IBM 650 and 705 programmers.

In 1964, two new, but highly similar tests--the Data Processing Aptitude Test (DPAT) and the Aptitude Test for Programmer Personnel (ATPP)--were developed by IBM to replace the PAT and RPAT (Hollenbeck & McNamara, 1965). The items contained in the DPAT and the ATPP are quite similar to those contained in their predecessors, the PAT and RPAT; and as their predecessors, the DPAT and the ATPP are paper-and-pencil tests of reasoning ability. They differ, however, in that the DPAT was designed specifically for selection and placement within IBM, whereas the ATPP was designed primarily to aid in the selection problems of IBM's customers. The DPAT is restricted to IBM internal use only, while the ATPP is available for use by IBM customers and schools. Published validation studies of these new tests, particularly the ATPP, are noticeably lacking. Currently, both tests are being widely used, however, as if they had been thoroughly validated.

The question has been raised as to whether further research in programming aptitude should be directed toward the development and validation of apparatus rather than paper-and-pencil type tests. Comparisons of the validities of these tests have been done in an attempt to answer this question. McNamara and Hughes (1961) compared the RPAT, a paper-and-pencil test of reasoning ability, with the LAD apparatus test. They found that when 20 IBM programmer trainees were administered the RPAT and the LAD, the LAD gave substantially the same results and apparently measured the same factors as the RPAT. A correlation of .82 was found between the 2 tests. In a similar study, Hollenbeck and McNamara (1965) compared the CUCPAT apparatus test with 2 paper-and-pencil tests of reasoning ability, the DPAT and the RPAT. One sample of 19 programmer trainees was administered the DPAT and the CUCPAT, while another sample of 27 was given the RPAT and the CUCPAT. The criterion variable was a rater's paired-comparison ranking of the programming performance of the trainees at the end of a 26-week training course. When the scores from each test were correlated with the criterion, Hollenbeck and McNamara found a <u>rho</u> of .54 for the RPAT as compared to a rho of .30 for the CUCPAT, and a rho of .54 for the RPAT as compared to a rho of only .06 for the CUCPAT. Hollenbeck and McNamara concluded that the CUCPAT did not offer better prediction of the criterion than the DPAT or RPAT.

Although the apparatus tests have high face validity, they are much more expensive and inconvenient to use than the paper-and-pencil tests. This inconvenience and expense could be justified if the apparatus tests were more accurate measures of programming aptitude than the paper-and-pencil tests. An interpretation of the research results cited above (Hollenbeck & McNamara, 1965; McNamara & Hughes, 1961) reveals, however, that the apparatus tests do not provide greater predictive validity of success in programming than the paper-and-pencil tests.

The purpose of the present study was to examine the validity of 2 recently developed paper-and-pencil tests of programmer aptitude in predicting success in college programming: the Computer Programmer Aptitude Battery (CPAB) (Palormo, 1967) and the ATPP (IBM, 1964). The 4 ATPP scores were used as predictor variables. These were scores for: Part I, a letter series subtest; Part II, a figure series subtest; Part III, an arithmetic subtest; and the Total ATPP. Similarly, the 6 CPAB scores were used as predictor variables. These scores were for: A Verbal Meaning subtest, a Reasoning subtest, a Letter Series subtest, a Number Ability subtest, a Diagramming subtest, and the Total CPAB.

The following relationships between the Total scores on the ATPP and the CPAB and final class grades in programming classes were hypothesized: (a) ATPP Total scores would be significantly related to grades; and (b) CPAB Total scores would be significantly related to grades.

In addition, an investigation was made of the predictive validity of each of the 3 ATPP and the 5 CPAB subtests taken one at a time. That is, first-order correlations were computed between each of the 8 subtests and the final class grades.

Finally, a step-wise multiple regression analysis was performed. This analysis was carried out in order to determine whether the ATPP and CPAB Total scores taken together would significantly increase the validity coefficient.

Method

Subjects

Fifty-seven programming students from a junior college in a southwestern metropolitan city served as _Ss_. Twenty-two of the _Ss_ in the sample were students enrolled in one of two sections of a FORTRAN programming course. Their ages ranged from 19 to 47, with a median age of 27 years. Nineteen were male; three were female. The remaining 35 Ss consisted of students enrolled in one of two sections of a COBOL programming course. Their ages ranged from 18 to 55 years, and the median age was 24 years. Twenty-eight were men, and seven were women. All Ss had completed high school. Four of the FORTRAN and three of the COBOL Ss held Bachelor's degrees. Only 2 of the FORTRAN and 3 of the COBOL Ss had had any prior computer programming experience. In each of these cases, previous experience was limited to a year or less.

Predictor Measures

The predictor variables included scores on the ATPP and CPAB. The ATPP consists of 3 separately timed subtests: 10 minutes for Part I, a 40 item letter series; 15 minutes for Part II, a 30 item figure series; and 30 minutes for Part III, a 25 item arithmetic reasoning series. When test-retest reliabilities were computed from the scores of 144 junior college students, the following coefficients were obtained: .79 for Part I, .73 for Part II, .57 for Part III, and .88 for the Total ATPP (IBM, 1964). In an industrial training program (IBM, 1964), correlations ranging from -.22 to .82 were found between the ATPP and grades in a number of 1401 and 1440 programming and Basic Computer Systems (BCS) courses. Significant average correlations were found for the 1401 classes (r=.45, p<.01), for the 1440 classes (r=.64, p<.01), and for the BCS classes (r=.56, p<.01).

The CPAB contains 5 individually timed subtests: 8 minutes for a Verbal Meaning subtest, 20 minutes for a Reasoning subtest, 10 minutes for a Letter Series subtest, 6 minutes for a Number Ability subtest, and 35 minutes for a Diagramming subtest. The Verbal Meaning subtest consists of 38 vocabulary words frequently used in the literature of systems engineering, mathematics, and business. Each vocabulary word is followed by 5 alternative words from which the S is to choose the 1 word whose meaning is most nearly the same as the vocabulary word. The Reasoning subtest consists of 24 word problems which must be transformed into mathematical symbols before a solution can be reached. The Letter Series subtest consists of 26 problems. In each problem, a series of letters are arranged according to a given pattern. The S's task is to discover the pattern and to choose, among 5 alternative letters, the 1 letter which would continue the pattern. The Number Ability subtest consists of 28 arithmetic problems designed to measure the S's ability to work quickly and accurately with numbers. The Diagramming subtest consists of 7 diagramming problems with flow charts. Five of the cells in the flow chart are incomplete. The _S_'s task is to select 1 statement from 5 alternative statements which correctly completes each flow chart cell. Palormo (SRA, 1967) reported that the reliability coefficients, computed by Kuder-Richardson Formula 20, for the Verbal Meaning, Reasoning, and Diagramming subtests were .86, .88, and .94, respectively. Split-half reliability coefficients of .67 for the Letter Series subtest and .85 for the Number Ability subtest were reported in the manual (SRA, 1967). The following ranges of correlations have been found between the CPAB and programming course grades in an industrial setting: .17-.34 for Verbal Meaning, .43-.52 for Reasoning, .09-.61 for Letter Series, .10-.61 for Number Ability, .25-.69 for Diagramming, and .46-.71 for the Total CPAB (SRA, 1967).

Criterion Measures

The performance criterion was the stanine equivalent of the final numerical class grade of each of the _Ss_ at the end of the semester. Criterion observations were made of the _Ss_ from 4 programming classes--2 COBOL and 2 FORTRAN classes. Instruction in both the FORTRAN and COBOL classes included an introduction to computer concepts and the fundamental principles of programming and programming techniques.

Both FORTRAN and COBOL represent high level computer languages which obviate the necessity of tedious direct memory assignment and numeric machine coding. In addition, both languages have been standardized so that they are machine-independent, i. e., they may be used on the equipment of any computer manufacturer. The name FORTRAN was coined from the first syllables of the words "Formula Translator." Similarly, the name COBOL was derived from the first letters in the phrase "Common Business Oriented Language." As the words "Formula Translator" imply, FORTRAN is an especially useful tool in the solution of scientific problems involving formulas, mathematical computations, or the manipulation of numerical variables. The COBOL language, as the phrase "Common Business Oriented Language" suggests, is particularly useful in the solution of business problems involving the English language. Even though FORTRAN is typically thought of as scientifically oriented and COBOL as business-oriented, they share many of the same attributes and are capable of solving many of the same problems. Consequently, both languages are currently being applied to business as well as scientific problems.

The Ss were graded on the bases of their numerical scores on class tests, programming performance, and a final examination. Two different FORTRAN instructors taught the 2 FORTRAN sections, but grades in both sections were based on identical class tests and assignments. One COBOL instructor taught the 2 COBOL sections. The same class tests and assignments were used as a bases for the grade in both COBOL sections. FORTRAN and COBOL Ss were given 7 programming problems on the System 360. Performance on each of these problems constituted about 7% of the grade. During the semester, FORTRAN and COBOL Ss were given 3 major tests. Performance on these tests accounted for 30% of the final grade. In the COBOL class, 10% of the grade was determined by the instructor's evaluation of homework assignments. On the last day of class, a multiple choice final examination was administered to the Ss in the FORTRAN and COBOL classes. The final examination accounted for 20% of the FORTRAN and for 10% of the COBOL grade. In order to place the FORTRAN and COBOL grades on a comparable scale, the final numerical grades were converted into stanines. Two separate stanine distributions were computed. One for the FORTRAN grades, and one for the COBOL grades.

Procedure

As part of a guidance program in data processing at the junior college, the counseling and guidance office had administered the ATPP to students over a period of 2 years. The ATPP scores of the *S*s participating in this study were obtained from this source. Administration of the CPAB took place in the programming classroom during a regular class period. The CPAB was administered to the *S*s according to the standard instructions and time limits enumerated in the manual.

At the end of the semester, the final numerical grades from the FORTRAN and the COBOL classes were obtained and transformed into their respective stanine distributions. By means of the product-moment correlation, the ATPP Total scores of all *S*s were correlated with the stanine transformation of their final class grades. The CPAB Total scores were correlated with the stanine equivalent of final grades. Each of the 3 ATPP and 5 CPAB subtests was correlated with the stanine transformation of final class grades. Using a step-wise regression technique, a multiple correlation was computed between the Total scores on the ATPP and CPAB and the stanine scores.

Results and Discussion

Separate means and standard deviations of the final numerical class grades were computed for the 22 FORTRAN *S*s, the 35 COBOL *S*s, and the entire sample of 57 *S*s.

These data, along with the means and standard deviations of the stanine equivalent of final class grades, are presented in Table 1.

TABLE 1

Means and Standard Deviations for Numerical Grades and Equivalent Stanines

Subjects	Numerical Grade		Stanine	
	M	SD	M	SD
FORTRAN (N=22)	85.55	8.55	5.00	2.09
COBOL (N=35)	67.06	26.31	5.00	1.99
FORTRAN and COBOL (N=57)	74.19	23.13	5.00	2.01

Pearson product-moment correlation coefficients between ATPP scores and the stanine equivalent of the final course grades for all 57 Ss in the sample are reported in Table 2. Inspection of Table 2 reveals that the ATPP Total scores, as well as the 3 ATPP

TABLE 2

Means, Standard Deviations, and Correlations of the *Aptitude Test for Programmer Personnel* with the Stanine Transformation of Final Course Grades for All 57 Programming Students

ATPP	*M*	*SD*	*r*
Part I	18.4035	8.2264	.4862**
Part II	13.0000	6.5165	.4883**
Part III	11.0702	6.5707	.5425**
Total Scores	42.4737	19.2902	.5571**

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

subtests, were significantly correlated at the .01 level with the stanine transformation of final grades. The hypothesis that ATPP Total scores would be significantly related to grades was, therefore, accepted.

The product-moment correlation coefficients between CPAB scores and grade stanines are presented in Table 3. As shown in Table 3, the correlation between the CPAB Total scores and class grade stanines also turned out to be significant at the .01 level. Thus, the hypothesis

TABLE 3

Means, Standard Deviations, and Correlations of the *Computer Programmer Aptitude Battery* with the Stanine Transformation of Final Grades for All 57 Programming Students

CPAB	*M*	*SD*	*r*
Verbal Meaning	13.4211	6.9564	.3757**
Reasoning	7.2807	4.9416	.3706**
Letter Series	9.2807	4.8209	.3337*
Number Ability	9.9825	4.5414	.3582**
Diagramming	18.5789	9.2330	.2802*
Total Scores	58.5439	25.1425	.4083**

*$p<.05$.

**$p<.01$.

that CPAB Total scores would be significantly related to grades was accepted. Table 3 also illustrates that only two of the CPAB subtest correlations--those for Letter Series and for Diagramming-- failed to reach significance at the .01 level. Both of these correlations were, however, significant at the .05 level.

TABLE 4

Results of the Step-Wise Multiple Regression Computed between Grade Stanines and Total Scores for the ATPP and CPAB

Predictor Variables	*R*	*F*
ATPP and CPAB Total Scores	.5581	.0890

In order to answer the question of whether the addition of the CPAB to the ATPP Total scores would significantly increase the validity coefficient, a step-wise multiple regression analysis was performed. Summary statistics for this analysis are shown in Table 4. The *F* value obtained in this analysis was low and nonsignificant ($p > .05$). Thus, according to these results, the combination of the ATPP and CPAB Total scores did not significantly improve prediction of the criterion.

From an examination of the data contained in Tables 2, 3, and 4, several conclusions were reached concerning the relationships between the criterion and the predictor variables for this sample of 57 programming students. First, the relationship between the criterion and each of the 10 predictor measures was statistically significant.

That is, each of the 4 ATPP and the 6 CPAB scores was significantly correlated with course grades. Second, the direction of the relationship between the criterion and each of the 10 predictor measures was positive. In other words, high scores on the ATPP and CPAB tended to be associated with high grades and low scores with low grades. Third, the correlations between the ATPP scores and class grades were higher than those between CPAB scores and class grades. Fourth, the ATPP Total scores correlated higher with final course grades than any other single measure in the study. Finally, using the ATPP and the CPAB Total scores together to predict programming grades did not significantly increase the predictive validity.

An interpretation of the results indicate that while the validity coefficients from both the ATPP and CPAB were statistically significant, the ATPP may be a more effective measure in predicting relative programming performance than the CPAB. In addition, there is apparently no advantage in using both measures together to predict programming success.

Although the primary objective of the present study was to investigate the predictive validity of the ATPP and CPAB for the entire sample of 57 programming students, an exploratory investigation was also made of the validity of these measures in predicting programming success for the 35 COBOL Ss as compared to the 22 FORTRAN Ss. In this

TABLE 5

Means, Standard Deviations, and Correlations of the *Aptitude Test for Programmer Personnel* with the Stanine Transformation of COBOL Grades

ATPP (*N*=35)	*M*	*SD*	*r*
Part I	16.4286	8.6748	.5277**
Part II	10.8000	6.8762	.7024**
Part III	8.9143	6.6967	.6615**
Total Score	36.1429	20.0970	.6885**

**p<.01.

investigation, data for FORTRAN and COBOL *S*s were analyzed separately. Pearson product-moment correlations between ATPP scores and the stanine equivalent of class grades were computed for the 35 COBOL *S*s (shown in Table 5) and for the 22 FORTRAN *S*s (shown in Table 6). Table 5 shows that all 4 of the ATPP correlations for the COBOL *S*s were significant at the .01 level. In fact, these correlations were actually higher than those found for the entire sample of 57 *S*s shown in Table 2.

TABLE 6

Means, Standard Deviations, and Correlations of the *Aptitude Test for Programming Personnel* with the Stanine Transformation of FORTRAN Grades

ATPP (N=22)	M	SD	r
Part I	21.5455	6.4640	.4963*
Part II	16.5000	3.9731	.1832
Part III	14.5000	4.7384	.4897*
Total Score	52.5455	12.8572	.4866*

*p<.05.

Correlation coefficients between the criterion and ATPP scores for the 22 FORTRAN Ss are reported in Table 6. The correlations for Part I, Part III, and Total ATPP scores were all significant at the .05 level, but Part II, the figure analogy series, was not significantly related to FORTRAN grades.

Comparison between Tables 5 and 6 shows that the correlations between the ATPP scores and programming grades were lower for the FORTRAN than the COBOL Ss. In order to ascertain whether the difference between the correlations of the ATPP Total scores for the COBOL ($\underline{r}$=.6885) and the FORTRAN $\underline{S}$s ($\underline{r}$=.4866) was statistically significant, the correlations were transformed into Fisher's $\underline{z}$ values and a test of the difference between two independent correlations was performed. According to the results of this test ($\underline{z}$=1.08, $\underline{p}$>.05), the correlations for the ATPP Total scores were not significantly different.

Further comparison between Tables 5 and 6 also reveals that of the 4 ATPP scores, Part II, the figure analogy series, correlated highest with COBOL grades but lowest with FORTRAN grades. A test of the difference between two independent correlations was computed for the Part II correlations ($\underline{r}$=.7024 for COBOL, and $\underline{r}$=.1832 for FORTRAN). The difference between the COBOL and FORTRAN correlations for Part II of the ATPP turned out to be significant ($\underline{z}$=2.37, $\underline{p}$<.05). An interpretation of this finding suggests that perhaps spatial reasoning ability is relatively more important for success in COBOL than in FORTRAN programming. However, further validation research is needed to determine whether this relationship will hold up with different samples.

Pearson product-moment correlation coefficients were computed between the criterion and the CPAB scores for the 35 COBOL $\underline{S}$s (shown in Table 7) and the 22 FORTRAN $\underline{S}$s

TABLE 7

Means, Standard Deviations, and Correlations of the *Computer Programming Aptitude Battery* with the Stanine Transformation of COBOL Grades

CPAB (N=35)	M	SD	r
Verbal Meaning	11.6000	6.4998	.4399**
Reasoning	5.2286	3.9563	.4494**
Letter Series	7.6857	3.8023	.4208*
Number Ability	8.4571	4.3001	.4307**
Diagramming	14.7143	8.8802	.4938**
Total Score	47.6857	21.8414	.5711**

*p<.05.

**p<.01.

(shown in Table 8). As can be seen in Table 7, all of the CPAB scores were positively related to COBOL grades. With the exception of the Letter Series subtest which was significant at the .05 level, all of the CPAB scores were correlated with COBOL grades at the .01 level of significance. An inspection of Table 7 reveals that the CPAB correlations for the 35 COBOL Ss were higher than those shown in Table 2 for the entire sample of 57 Ss. A similar observation was pointed to earlier regarding the ATPP correlations shown in Tables 2 and 5.

TABLE 8

Means, Standard Deviations, and Correlations of the Computer Programmer Aptitude Battery with the Stanine Transformation of FORTRAN Grades

CPAB (N=22)	M	SD	r
Verbal Meaning	16.3182	6.8061	.3376
Reasoning	10.5455	4.6468	.4211
Letter Series	11.8182	5.2521	.3162
Number Ability	12.4091	3.8749	.3405
Diagramming	24.7273	5.9296	-.0192
Total Score	75.8182	20.1060	.3542

When the CPAB scores of the 22 FORTRAN Ss were correlated with the criterion variable, low and nonsignificant validity coefficients resulted (shown in Table 8). These correlations, however, might have reached significance had the sample been larger. According to the results, the

CPAB was significant in predicting programming performance in the COBOL but not in the FORTRAN course. However, when a test of the difference between two independent correlations was computed for the correlations of the CPAB Total scores (r=.5711 for COBOL, and r=.3542 for FORTRAN), a non-significant difference resulted (z=.964, p>.05). Thus, additional validation studies are needed to determine if the validity coefficients shown in Tables 7 and 8 will hold up with new samples of COBOL and FORTRAN _S_s.

From an examination of the data summarized in Tables 5, 6, 7, and 8, several conclusions were reached concerning the relationship between the criterion and the ATPP and CPAB scores for the 35 COBOL _S_s as compared to the 22 FORTRAN _S_s. First, all 4 of the ATPP scores were significantly related to COBOL grades, and with the exception of Part II, all of the ATPP scores were significantly related to FORTRAN grades. An interpretation of this result suggests that the ATPP measures characteristics which are important in both COBOL and FORTRAN programming. Second, all 6 of the CPAB scores were significantly related to COBOL grades, but none of the CPAB scores were found to be significantly related to FORTRAN grades. This finding indicates that the CPAB may be measuring characteristics which are important in COBOL programming but relatively unimportant in FORTRAN programming. Further validation research is recommended to substantiate this finding. Finally, grades for both the COBOL and the FORTRAN Ss correlated higher with ATPP than CPAB scores.

Summary and Conclusions

The purpose of the present study was to examine the validity of two paper-and-pencil tests, the ATPP and the CPAB, in predicting success in computer programming. A total of 57 programming students of both sexes enrolled in a junior college in a large southwestern city served as Ss. Twenty-two of the Ss in the sample were FORTRAN students; thirty-five were COBOL students. The criterion of performance in 2 FORTRAN and 2 COBOL programming classes consisted of the final class grade based on the student's numerical scores on class tests and programming performance. The 4 ATPP and 6 CPAB scores served as predictor measures. All Ss were administered the ATPP and CPAB, and the FORTRAN and COBOL grades were placed on a comparable scale by converting the final numerical class grades into stanines.

The method of study involved correlating the scores of each of the 4 ATPP and 6 CPAB predictor measures with the stanine transformation of grades using the Pearson product-moment correlation. In addition, a step-wise multiple regression analysis was performed to determine whether the ATPP and CPAB Total scores taken together would significantly increase the validity coefficient.

An analysis of the results obtained with all 57 Ss revealed a significant positive relationship between the programming grades and scores from each of the 4 ATPP and 6 CPAB predictor measures. The hypothesis that a significant relationship between the ATPP Total scores and final class grades would exist was accepted, as was the hypothesis that there would be a significant relationship between the CPAB Total scores and class grades. Both of these correlations were significant at the .01 level. Further examination of the results revealed that the ATPP tended to yield higher correlations with class grades than the CPAB, and that the ATPP Total scores correlated higher with the criterion than any other single predictor measure. When the step-wise multiple regression analysis was performed to determine whether the addition of the CPAB to the ATPP Total scores would significantly increase the predictive validity, a low and nonsignificant F value resulted ($p > .05$). Thus, the combination of the ATPP and CPAB Total scores did not significantly improve prediction of the criterion.

Although the primary objective of the present study was to investigate the predictive validity of the ATPP and CPAB for the entire sample of 57 Ss, data for the 22 FORTRAN and 35 COBOL Ss were also analyzed separately. This analysis consisted of computing separate Pearson product-moment correlations between the criterion and the predictor measures for the FORTRAN and the COBOL Ss.

An inspection of the results obtained from the separate analysis of the COBOL and FORTRAN data showed that each of the 4 ATPP scores was significantly related to COBOL grades, and except for Part II, each of the ATPP scores was significantly related to the FORTRAN grades. Furthermore, correlations between COBOL grades and each of the 6 CPAB scores were also significant though they tended to be lower than those between ATPP scores and COBOL grades. None of the CPAB scores, however, were significantly related to FORTRAN grades.

Comparison of the data from the separate analysis of the 35 COBOL and 22 FORTRAN Ss with the data obtained from the entire sample of 57 Ss revealed that the ATPP consistently yielded higher correlations with programing grades than the CPAB, and that the ATPP Total scores provided the most consistent single predictor of programming success in the study. The ATPP validity coefficients obtained in the present study were comparable to those found in previous research with the

PAT, the IBM paper-and-pencil test of reasoning ability which preceeded the ATPP (McNamara & Hughes, 1961). These results tend to support the applicability of the ATPP across different programming situations. They lend further support to the importance of reasoning ability in programming performance and to the practicality of the refinement of such paper-and-pencil tests of programming aptitude. The finding that the ATPP tended to yield higher validity coefficients than the CPAB is noteworthy since the ATPP is a shorter test and takes less time to administer than the CPAB.

Although the CPAB was significant in predicting the programming performance of the COBOL Ss as well as the performance of the entire sample of 57 Ss, the failure to obtain significant correlations between the CPAB and FORTRAN grades raises the question of the applicability of the CPAB to various types of programming. Moreover, an analysis of these results revealed that by far the largest portion of the variance in the correlations between the CPAB scores and the programming grades for all 57 Ss was accounted for by COBOL grades. Cross-validation research is, therefore, needed to determine whether the applicability of the CPAB in predicting relative performance is limited only to certain kinds of programming.

REFERENCES

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

Boyce, R. W. The prediction of achievement in college algebra. *Educational and Psychological Measurement*, 1964, *24*, 419-420.

Distefano, M. K., Jr., & Rice, M. L. Predicting academic performance in a small southern college. *Educational and Psychological Measurement*, 1966, *26*, 487-488.

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.

International Business Machines Corporation. *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: an electronically operated test. Behavioral Science, 1957, 2, 291-300.

Katz, A. Prediction of success in automatic data processing programming course. (USA PRO OCRD Technical Research Note No. 126), 1962.

Langmuir, C. A logical machine for measuring problem solving ability. Proceedings of the Eastern Joint Computer Conference. New York: Association for Computing Machinery, 1960, 18, 1-9.

Lavin, D. E. The prediction of academic performance. New York: Russell Sage Foundation, 1965.

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

Opler, A. Testing programming aptitude. Datamation, 1963, 9, 28-31.

Palormo, J. M. Computer Programmer Aptitude Battery (Manual). Chicago: Science Research Associates, Inc., 1967.

Peres, S. H., & Arnold, P. H. Identifying programmer behavior. Datamation, 1963, 9, 40.

Perry, D. K. Programmer selection procedures. Field Note-6900. Santa Monica, Calif.: System Development Corp., 1962.

Perry, D. K. Training and job performance validities of programmer trainee selection variables. Memorandum TM-2172. Santa Monica, Calif.: System Development Corporation, 1964.

Perry, D. K. Vocational interests and success of computer programmers. Personnel Psychology, 1967, 50, 517-524.

Perry, D. K., & Cannon, W. M. Project summary: vocational interests of computer programmers. Memorandum TM-2655. Santa Monica, Calif.: System Development Corporation, 1965.

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

Perry, D. K., & Cannon, W. M. Vocational interests of female computer programmers. Journal of Applied Psychology, 1968, 52, 31-35.

Perry, D. K., & Cantley, G. Computer programmer selection and training in System Development Corporation. Memorandum TR-2234. Santa Monica, Calif.: System Development Corporation, 1965.

Roemmich, H. Testing programmer efficiency. Journal of Data Management, 1963, 1, 24-26.

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

Sweetland, A. Factors in selecting and training programmers. Memorandum RM-3245-PR. Santa Monica, Calif.: Rand Corporation, 1962.

Watson, J. Programmer selection survey. Data Processing, 1961, 3, 9-13.