Identifying Predictors of Programming Achievement

YAVUZ ERDOGAN
Department of Computer Education and Instructional Technology
Marmara University,
Goztepe Campus, Kadikoy, Istanbul
TURKEY

EMİN AYDIN
Department of Mathematics Education
Marmara University,
Goztepe Campus, Kadikoy, Istanbul

TOLGA KABACA
Department of Mathematics
Usak University
Usak, TURKEY

Abstract: The main purpose of this study is to explore the predictors of programming achievement. With this aim in mind, the students' achievement in the programming courses is specified as the dependent variable and computer achievement, creativity, problem solving, general aptitudes, computer attitudes and mathematics achievement are specified as the independent variables. A correlational design was used to explain the relations between dependent and independent variables. The study group consists of 48 high school students in Profilo Anatolia Technical High School, Istanbul. At the end of the study, significant relations were found between the students' programming achievement and their computer achievement, general aptitudes and mathematics achievement. Also, in order to determine the predictors of the students' programming achievement, multiple regression analysis was applied. The findings reveal that only one variable that significantly predicts the students' programming achievement is general aptitude.

Key-Words: Programming achievement, creativity, problem solving, general aptitudes, computer attitude, mathematics achievement

1 Introduction

There have been many studies in recent years into academic achievement in computer programming [1, 2, 3, 4]. Today, industry is keen to accept as many graduates as the academic institutions can produce, and there is an assumption that any bright student can be successful in computer programming. However, experience in the classroom suggests that this might not be the case. Students who are proficient in many other subjects sometimes perform poorly in programming [2].

The developments in programming languages and methods, and their teaching have up to now hardly been linked to a psychological study of the activity of programming. Psychology must go beyond the procedural aspect of programming; because it is becoming more and more important nowadays due to the variety of applications and the training that programmers receive [5]. Prior research indicates that standardized measures of aptitude (e.g. SAT and ACT scores), prior academic performance (e.g. high school GPA) and effort or motivation explain a significant portion of the variation in student performance [6, 7, 8].

In a review of studies attempting to predict programming achievement done up to 1990, Hostetler and Corman make a specific case for the inclusion of cognitive factors in any study of this kind [7, 9]. They found that some of the demographic, academic, computer exposure or cognitive variables were particularly strong predictors of class performance. According to Taylor and Mounfield [10] prior experience in programming provides a significant predictor of how students perform in the programming courses. They found that prior exposure whether at the high school or college level is an important factor in students' performance in computer programming.

The link between mathematics ability and programming is widely evidenced in the related literature. Several of the reviewed studies showed that achievement in mathematics is a good predictor of achievement in computer science [2, 4, 11, 12, 13]. There is a belief that the concepts which a student has to comprehend in order to master mathematics problems are similar to those for programming [2, 11].

There appears to be a number of other factors which influence programming performance. In general the reviewed research found correlation between computer attitudes and computer programming [14, 15]. Earlier studies also indicated that demographic data has an impact on programming performance [2, 8, 16]. In [8] five factors were reviewed as potentially predictive to performance in programming which are problem solving ability, motivation, learning style, previous experience, and gender. Even though these variables are helpful in predicting performance in computer programming, it appears that they could also predict performance in other fields. These findings reveal that programming ability is different from other skills.

Considering all these points, the current study hopes to explore the correlations between students' programming achievement and their computer achievement, creativity, problem solving ability, general aptitudes, computer attitude, and mathematics achievement. With this aim in mind, research questions can be stated as follows;

1. Is there a significant relationship between the students' performance in programming and their achievement in a computer course not involving programming.
2. Is there a significant relationship between the students' performance in programming and their creativity?
3. Is there a significant relationship between the students' performance in programming and their problem solving ability?
4. Is there a significant relationship between the students' performance in programming and their general aptitudes?
5. Is there a significant relationship between the students' performance in programming and their computer attitude?
6. Is there a significant relationship between the students' performance in programming and their mathematics achievement?
7. Are there any mental factors that significantly predict the students' programming achievement? If so, what are they?

2 Methodology

2.1 Research Method

In the current study, a correlational design was used to investigate relations between students' programming achievement and their computer achievement, creativity, problem solving ability, general aptitudes, computer attitude, and mathematics achievement.

2.2 Participants

The study was conducted in 2005-2006 education year and the sample of this study consisted of 48 students from a technical high school in Istanbul. This school aims particularly to equip the students with computer and educational technologies. The students are capable of computer technologies and programming languages. 25% of the sample was female while 75% of them were male.

2.3 Instruments

In this study five different measurement tools were used which are programming achievement test, KAI creativity scale, problem solving inventory, general skills test battery, and computer attitude scale. Students' scores in the 'Introduction to Computers' course were taken as a measure for the computer scores. This course is a two hours/week course who aims to teach basic skills a beginner might need for using computers which involves information about how to use some Microsoft Office programs (i.e. Word, Excel and Power Point) in a Windows environment. Similarly, students grades from their previous mathematics courses in the school were used as an indicator of their mathematics performance. Data collection tools are explained below in detail.

2.3.1 Programming Achievement Test (PAT)

A multiple-choice test consisting of 25 questions was developed by the researchers in order to measure students' performance in programming. Validity and reliability studies of the PAT were carried out again by the researchers. After the item analysis 4 items were removed from the test. At the end of the reliability and validity analysis administered with the remaining 21 questions, the cronbach alfa internal consistency was found to be 0.72.

2.3.2 KAI Creativity Scale

KAI Creativity Scale [18] contains 33 questions. Students were asked to respond to the statements using a five-point Likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). Validity and reliability studies were carried out by the same researchers, and the cronbach alfa internal consistency was found to be 0.89.

2.3.3 Problem Solving Inventory (PSI)

This scale was designed by [19] which is a five-point likert scale ranging from 1 (Strongly Disagree) to 5 (Strongly Agree). PSI measures the attitudes of the students about problem solving and consisted of 38 items. 3 items were removed from the scale as their item total-item remainder correlations were insufficiently low. The latest version of the scale was found to have an internal consistency coefficient of 0.86.

2.3.4 General Skills Test Battery (GSTB)

The original form is in French and it has been adapted to Turkish by [20]. The test measures analytic thinking, abstract thinking and spatial perception. It's a performance test and can be applied to individuals between 15-17 years of age. The test battery containing 113 items has three different dimensions which are; Letter Series (25 items), Shape Recognition (48 items) and Volume Surface Expansion (40 items) [21]. The total score from the three different tests of the battery constitutes the students' general skills. Validity and reliability study of the GSTB was carried out again by the researchers and cronbach alfa coefficient was found as 0.85 for Letter Series; 0.94 for Shape Recognition and 0.84 for Volume Surface Expansion.

2.3.5 Computer Attitude Scale (CAS)

Computer attitudes of the students were measured using the Computer Attitudes Scale [22]. This 42-item scale asks participants how frequently they agree with statements such as "Studying with computers is entertaining", "Computers make me angry", and "I believe that computers are beneficial". Participants rated how strongly they agree or disagree with each statement on a five-point scale. Higher scores indicate the greater levels of computer attitudes. There was high internal reliability for this scale; the standardized item alpha was 0.88.

2.4 Data Analysis

A bivariate Pearson's correlation was applied between the students' programming achievement and their computer achievement, creativity, problem solving ability, general aptitudes, computer attitude, and mathematics achievement. In order to obtain the most suitable regression equivalent in explaining the students' programming achievement, multiple regression analysis was used. Multiple regression analysis is used in interpreting the total variance of the dependent variable explained by the independent variables and its statistical significance. Programming achievement was included as the dependent variable and other factor as the independent variables.

The significance level for all the statistical results in the study was accepted to be 0.05 and all the results were tested two-ways. For statistical analysis the software used was SPSS 13.0.

3 Results

For the data analysis, first, descriptive statistics were presented. Then, the correlations between the students' programming achievement and other variables were presented. Lastly, results from the multiple regression analysis were stated.

Table 1. Descriptive Statistics

Variables	Mean	Std.Dev.	Std.Er.
Programming achievement	66.41	13.28	1.44
Computer achievement	65.64	14.03	1.73
Creativity	119.08	10.39	1.50
Problem solving	132.04	13.31	1.92
General aptitudes	52.57	19.60	1.82
Computer attitude	172.68	20.68	2.98
Math. achievement	47.66	17.81	2.57

Table 1 shows the descriptive statistics of the measurements. The mean is 66.41 for programming achievement; 65.64 for computer achievement; 119.08 for creativity; 132.04 for problem solving; 52.57 for general aptitudes; 172.68 for computer attitude, and 47.66 for mathematics achievement.

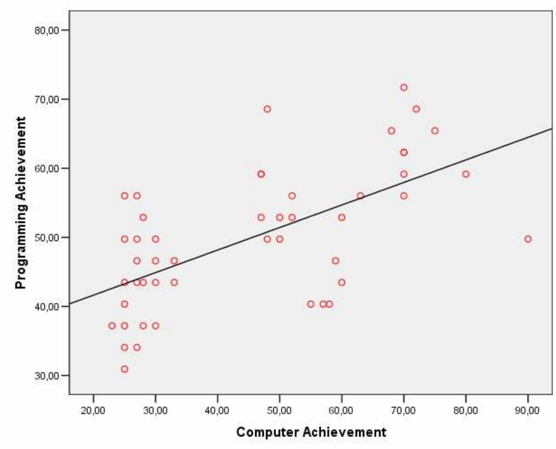
Table 2. The correlation values between programming achievement and independent variables

	Programming achievement	
	r	P
Computer achievement	0.621	.000
Creativity	0.053	.720
Problem solving	0.072	.626
General aptitudes	0.934	.000
Computer attitude	0.106	.474
Math. achievement	0.447	.001

A bivariate Pearson's correlation coefficients were run to determine the degree of relationship between the students' programming achievement and their computer achievement, creativity, problem solving ability, general aptitudes, computer attitude, and mathematics achievement. Computer achievement, general aptitudes and mathematics achievement had significant correlations with programming achievement at the .01 level (see Table 1). However, there was no significant correlation between programming achievement and creativity, problem solving ability, and computer attitude.

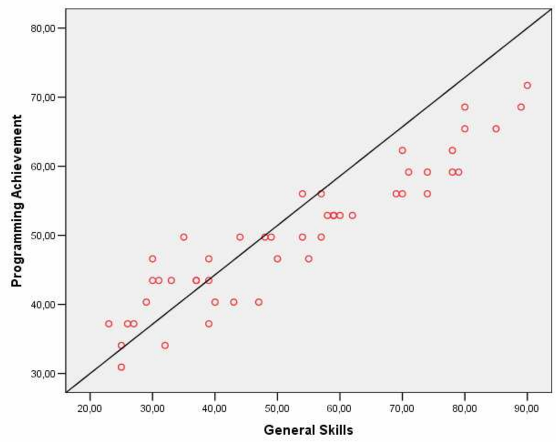
A strong correlation was found, as expected, between students' programming achievement and their computer achievement scores ($r=0.621$; $p<0.01$). According to this finding, 38.5% of the variation in students' programming achievement was explained by computer scores ($r\text{-square}_{\text{effect size}}=0.385$). As a result of this finding, it can be said that the students who are successful in computer courses are also successful in programming courses. There is a linear correlation between the performance in the programming and computer achievements (see Figure 1).

Figure 1. Scatter plot for the programming achievement and computer achievement



On the other hand, the strongest correlation score was detected between the students' programming achievement and their general aptitudes ($r=0.934$; $p<0.01$). This result indicates that standardized measures of general aptitude scores explain 87.2% of the variation in student performance ($r\text{-square}_{\text{effect size}}=0.872$). This finding reveals that general aptitude is an important factor to students' performance in computer programming. The perfect linear correlation between students' programming achievement and their general aptitudes is presented in Figure 2.

Figure 2. Scatter plot for the programming achievement and general aptitudes



The second highest correlation was found between students' programming achievement and mathematics achievement ($r=0.447$; $p<0.01$). According to this finding, 19.8% of the variation in students' programming achievement was explained by mathematic scores ($r\text{-square}_{\text{effect size}}=0.198$). Mathematics, being a demonstrative discipline, tries to demonstrate numbers, shapes and the relations between these by analyzing them.. Mathematics does this through reasoning which is the common point between programming and mathematics. This finding supports the belief that the concepts which a student has to comprehend in order to master mathematics problems are similar to those for programming. The linear correlation between students' programming achievement and their mathematics achievement is presented in Figure 3.

Figure 3. Scatter plot for the programming achievement and mathematics achievement

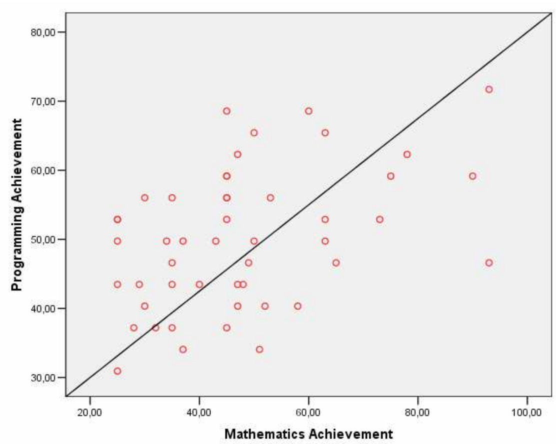


Table 3. The results of the multiple regression analysis: The predictors of programming achievement

Variables	B	βeta	T	P
Constant	31.99		3.99	.001
Computer achievement	0.04	0.07	0.88	.387
Creativity	-0.08	-0.09	-1.48	.148
Problem solving	0.06	0.08	1.35	.182
General aptitudes	0.45	0.90	12.08	.000
Computer attitude	-0.03	-0.06	-1.04	.304
Math. achievement	-0.01	-0.02	-0.29	.773

R=0.940 R^2=0.884; F=52.221 p=0.000

In order to identify the predictors of the students' programming achievement, multiple regression analysis was applied. As a result, positive relations were detected between the students' programming achievement and independent variables such as computer achievement, creativity, problem solving, general aptitudes, computer attitude and mathematics achievement (F=52.221; p<0.01). Independent variables explain about 88.4% of the total variance of the programming achievement.

The only variable that significantly predict students' programming achievement is general aptitudes (t=12.08; p<.01). This result reveals that general aptitudes provide a significant indication of how students perform in the programming courses. The regression equation for predicting the students' programming achievement is presented in below:

Programming Achievement = 31.99 + 0.45 general aptitudes + 0.06 problem solving + 0.04 computer achievement - 0.01 mathematics achievement - 0.03 computer attitude - 0.08 creativity

4 Discussions

the study which investigates the factors influencing their programming performance reveals some noteworthy findings. For example, there is a significant correlation between programming and non-programming computer performances (r=0.621; p<0.01). It is quite probable that students' knowledge about computers makes it easier for them to perform better in programming. This supports finding of the earlier studies [2, 8, 10].

Other main factor is the genaral ability. The analysis of results clearly suggests a very high impact of genaral ability on programming performance (r=0.934; p<0.01). Ability tests can assess skills on cognitive, verbal, spatial and psyco-motor domains including individuals' powers of comprehension, abstarct thnking skills and space perceptions [20, 21].

It is not surprising that students with high genearl ability scores perform better in tasks involving computer programming, as it is a skill which necesitates high amount of abstarction capacity in the performer.

The results of regression analysis indicates that student' genaral ability scores are reliable predictors of their programming performances (t=12.083; p<0.01) which supported finding of the earlier studies [15, 23].

In this study, it has been revealed that there is a significant correlation between the students' performance in the programming courses and their mathematics achievement at the level of 0.01. (r=0.447; p<0.01). These results support the theories and researches to date. Several of the reviewed studies showed that performance in mathematics was a good predictor of performance in computer programming [2, 6, 12, 24]. There is a belief that the concepts which a student has to comprehend in order to master mathematics problems are similar to those for programming [11]. This could be that computing as a subject requires a structure and approach with which students have some experience, and similar cognitive skills used in the study of mathematics. Mathematics aptitude is thus often a pre-requisite for acceptance into computer science programs [2].

On the other hand the study has found no correlation between programming achievement and creativity, problem solving, and computer attitude.

5 Future Directions

There are several limitations of this study. The participants were from a technical high school. Further research could include participants from other institutions like universities. Moreover, additional researches are needed in order to investigate the other factors that might influence programming performance. This study excluded factors such as personal traits, learning styles and demographic factors. These factors could be included in future research.

References:

[1] McNamarah, S. & Pyne, R. Teaching a First Level Programming Course: Strategies for Improving Students Performance, *Journal of Art Science & Technology*, Vol.1, p.42–49, 2004.

[2] Byrne, P. & Lyons, G. The Effect of Student Attributes on Success in Programming, *Association for Computer Machinery*, Vol.36, No.3, p.49-52, 2001.

[3] Begum, M. An Ontology for Teaching Programming *Association for Computer Machinery*, Vol.43, p.169-182, 2003.

[4] Fowler, L., Campbell, V., McGill, D. & Roy, G. An Innovative Approach to Teaching First Year

Programming Supported by Learning Style Investigation, Paper presented at the Australasian Association for Engineering Education, Melbourne, 2002.

[5] Hoc, J., Green, T.G., Samurcay, R. & Gilmore, D.J. *Psychology of Programming*, London: Academic Press, 1990.

[6] Eskew, R.K. & Faley, R.H. Some Determinants of Student's Performance in the First College-Level Financial Accounting Course. *The Accounting Review.* Vol.11, No.1, p.137-147, 1981.

[7] Hostetler, T. R. Predicting Student Success in an Introductory Programming Course, *ACM SIGCSE Bulletin*, Vol. 15, No. 3, p.40-49.

[8] Goold, A. & Rimmer, R. Factors Affecting Performance in First Year Programming, *ACM CIGCSE Bulletin*, Vol. 32, p.39-43, 2000.

[9] Corman, L, S. Cognitive Style, Personality Type, and Learning Ability as Factors in Predicting the Beginning Programming Student, *Association for Computer Machinery,* Vol.18, p.80-89, 1986.

[10] Taylor, H. G. & Mounfield, L. C. The Effect of High School Computer Science, Gender, and Work on Success in College Computer Science, ACM SIGCSE Bulletin 21.1, p.195-198, 1989.

[11] Werth, L. H. Predicting Student Performance in a Beginning Computer Science Class, In Proceedings of the Seventeenth SIGCSE Technical Symposium on Computer Science Education, Cincinnati, Ohio, United States, p.138-143, 1986.

[12] Campbell. P. F. & McCabe, G. P. Predicting the Success of Freshmen in a Computer Science Major, *Communications of the ACM*, Vol.27, No.11, p.1108-1113, 1984.

[13] Chmura, G. A. What Abilities Are Necessary for Success in Computer Science?, *ACM SIGCSE Bulletin*, Vol.30, No.4, p.55-58, 1998.

[14] Dey, S. & Mand, L. R. Effects of Mathematics Preparation and Prior Language Exposure on Perceived Performance in Introductory Computer Science Courses, CM SIGCSE Bulletin, Vol.18, No.1, p.144-148, 1986.

[15] Austin, H. S. Predictors of Pascal Programming Achievement For Community College Students, Proceedings of the eighteenth SIGCSE technical symposium on Computer science education, Missouri, United States, p.161-164 , 1987.

[16] Grant, N. A Study on Critical Thinking, Cognitive Learning Style, and Gender in Various Information Science Programming Classes, *Association for Computer Machinery*, Vol.42, s.96-99, 2003.

[17] Ramalingam,V. & LaBelle, D., Wiedenbeck, S. Self-Efficacy and Mental Models in Learning to Program, The 9th Annual SIGCSE Conference on Innovation and Technology In Computer Science, Leeds, United Kingdom. p.171-175, 2004.

[18] Topaktas, A. *Meslek Lisesi Öğrencilerinin Yaratıcılık Düzeylerinin Değerlendirilmesi*. Marmara University, Science Institute (Unpublished Thesis), Istanbul, 2001.

[19] Kasap, Z. *İlkokul 4. Sınıf Öğrencilerinin Sosyo-Ekonomik Düzeye Göre Problem Çözme Başarısı İle Problem Çözme Tutumu Arasındaki İlişki*, Marmara University, Social Science Institute (Unpublished Thesis), Istanbul, 1997.

[20] Ozcan, A. *Ülkemiz İçin İsabetli Olabilecek Bir Mesleğe Yöneltme Denemesi*, Publishing of Literature Faculty, Istanbul University, Istanbul, 1985.

[21] Oner, N. (1996). *Türkiye'de Kullanılan Psikolojik Testler,* Publishing of Bogazici University, No.584, Istanbul.

[22] Deniz, L. *Bilgisayar Tutum Ölçeği BTÖ-M'nin Geçerlilik, Güvenirlilik, Norm Çalışması ve Örnek Bir Uygulama*. Marmara University, Social Science Institute (Unpublished Dissertation), Istanbul, 1994.

[23] Alspaugh, C.A. Identification of Some Components of Computer Programming Aptitude, *Journal for Research in Mathematics Education*, Vol.3, No.2, s.92-98, 1972.

[24] Konvalina, J., Wileman, S. S. & Stephens, L. J. Math Proficiency: A key to Success for Computer Science Students, *Communications of the ACM*, Vol.26, No.5, p.377-382, 1983.