International Journal on New Trends in Education and Their Implications
July 2014 Volume: 5 Issue: 3 Article: 15 ISSN 1309-6249

PREDICTORS OF C# PROGRAMMING LANGUAGE SELF EFFICACY AMONG VOCATIONAL COLLEGE STUDENTS

Dr. Harun ÇİĞDEM
First Lieutenant at Noncommissioned Officer School
Balıkesir- TURKEY

Osman Gazi YILDIRIM
First Lieutenant at Noncommissioned Officer School
Balıkesir- TURKEY

ABSTRACT

The purpose of this descriptive-correlational study is to examine the relationship between C# programming self-efficacy and students' age, type of graduated high school, experience of computer usage, frequency of computer use, and programming courses experience. A scale with twenty-eight items assessing C# programming self-efficacy was adapted from Ramalingam and Wiedenbeck's the computer programming self-efficacy scale. The scale was utilized at the end of the Visual Programming (C#) course via learning management system with a questionnaire about demographics and computer experiences. One hundred and sixteen college students from Computer Technologies Department and Electronic Communication Technologies Department participated in the study. Pearson product correlation, regression analysis and t-tests were utilized to analyze the resulting data. Results indicates that C# Programming self-efficacy has no significant relationship with each of the students' age, type of graduated high school, departments and frequency of computer use. It was additionally obtained that the prior programming course experience and computer use experience in years are predictors of C# programming self-efficacy. C# programming self-efficacy of students who has taken computer programming course before is significantly stronger than students who haven't taken any computer programming course previously. Understanding students' self-efficacy beliefs about computers programming is useful to design effective programming courses.

Key Words: Programming course, C# programming language, vocational college, self-efficacy.

INTRODUCTION

Educational researchers accepted that improving students' self-beliefs about their academic capabilities play an important role in improving their academic performance. Researchers have assessed self-beliefs in a more task-specific way, and focusing on self-efficacy was one of the most important of these efforts (Zimmerman, 2000). Self-efficacy is "*people's judgments of their capabilities to organize and execute courses of action required attaining designated types of performance*" (Bandura 1986, p. 391) and individual's self-efficacy is an important factor in performance over a wide range of situations (Bandura 1977, 1986) and it is especially important if the situation relates to education (Askar & Davenport, 2009).

Self-efficacy influences personal selection of strategies, the level of effort shown, the level of persistence in difficulties, and performance outcomes while solving problems (Bandura 1986; Zimmerman 1995). Individuals with higher self-efficacy beliefs perceive themselves as capable of performing certain tasks or activities however individuals with low self-efficacy beliefs perceive themselves as less capable and less likely to attempt

at these certain tasks or activities (Askar & Davenport, 2009; Aşkar & Umay, 2001; Bandura, Adams, & Beyer, 1977; Wiedenbeck, 2005).

According to Bandura (1977, 1986), four sources of information: the individual's experience related to the skill, experiences of observing the performance of others similar to oneself, verbal persuasion, and physiological reactions -fear, stress, also fatigue, aches & pains- that people use partly to judge their capacities, affect individuals' self-efficacy beliefs. Self-efficacy measures focus on respondents' qualifications to achieve given task demands, not their personalities or their feelings about themselves in general (Zimmerman, 2000). This is also applicable to computer programming domain - how students consider themselves to perform while doing computer programming tasks-projects. Researches related to computer self-efficacy has been studied, however fewer researches related to computer programming self-efficacy has been done.

Ramalingam, La Belle, and Wiedenbeck (2004) explored the effects of students' self-efficacy and mental models of programming on learning to program. They found that previous programming experience influence programming self-efficacy. In another study, Askar and Davenport (2009) identified variables that are related to engineering students' Java programming self-efficacy in Turkey, concluding with gender, computer experience, and family usage of computers factors. They found that self-efficacy of students influenced by computer experience and computer skills, and also they found students' gender and family usage of computers didn't affect students' self-efficacy. In a Nigerian University, Jegede (2009) investigated engineering students' Java programming self-efficacy related with students' programming experience. The findings from the study showed that there is significant relationship between students' Java programming self-efficacy and each of the computer use and programming experience factors. Taking programming course and weighed scores of programming courses were found as predictors of Java Programming self-efficacy.

Studies identifying distinct factors affecting programming self-efficacy of vocational college students are absent. In this context, recognizing these factors is important in order to teach students computer programming skills effectively.

This study investigates relationship between C# self-efficacy and each of students' age, type of graduated high school, experience of computer usage in years, frequency of computer use, and programming courses experience. This study will answer the following questions:

1. Is there a significant difference between Computer Technologies students and Electronic Communication Technologies students' C# programming self-efficacy?
2. How do students' age, type of graduated high school, experience of computer usage in years, frequency of computer use, and programming courses experience affect C# programming self-efficacy beliefs?

METHOD

Participants, research instruments, data collection and method of analysis are described in this section.

Participants

Data were collected across a vocational college in Balıkesir, Turkey, in January, 2014. Study participants consisted of 116 students enrolled in "Visual Programming (C#)" course. Twenty-nine students were from Computer Technologies Department and other students were from Electronic-Communication Technologies Department. Visual programming is a must course for all students from two departments. According to the innate nature of school, majority of the students were male, which can be considered as a limitation for the entire study. Participants' demographics data were as in the Table 1.

Table 1: Participants' *Demographics*

		n	Percentages (%)
Gender	Male	114	99.3
	Female	2	0.7
Ages	18	13	11.2
	19	31	26.7
	20	45	38.8
	21	21	18.1
	22	5	4.3
	23	1	.9
Types of Graduated High School	Computer-Electronic	84	72.4
	Other Departments	32	27.6
Departments	Computer	29	25.0
	Electronic	37	75.0
Programming Course Experience	Yes	29	25.0
	No	87	75.0

All subjects were asked to respond to the instrument and their responses were guaranteed confidentiality and they were told that the data gathered would only be used for academic purposes. All of 116 students filled out the questionnaire. Students who are from ECT are sophomores and students who are from CT are freshmen.

Research Instruments

Students completed an online questionnaire with two sections through learning management system at the end of the semester. The first section was related to students' demographic/personal data including age, department, experience of computer usage in years, frequency of computer use and programming course experience.

The second section of the instrument was C# programming self-efficacy scale. It includes 28 items with seven-point Likert scale. Items of C# programming self-efficacy scale were adapted from the computer programming self-efficacy scale of Ramalingam and Wiedenbeck (1998), such as: "I can write syntactically correct C# statements."; "I can write a C# program that computes the average of any given number of values.", "I can develop my own C# applications.", "I could come up with a suitable strategy for a given programming project in a short time." The items are presented positively worded statements and the scale, which was in Turkish, is given in Appendix 1. The items were coded from 7 (absolutely confident) to 1 (mostly not confident). A higher score indicated higher C# programming self-efficacy. Maximum score that can be obtained from the scale was 196 while the minimum was 28. In order to determine the internal reliability of the scale, researchers performed a reliability analysis with the use of Cronbach's alpha after the data collection phase by using SPSS 20.0 computer software. The reliability of the scale was 0.97.

Data analysis procedures

The online questionnaire was utilized at the end of 2013/2014 academic year fall semester. The SPSS statistical package program was used to analyze the data using descriptive statistics, independent samples t-test, Pearson product correlation and regression analysis.

FINDINGS

Descriptive statistics

C# Programming Self-Efficacy items' means and standard deviations are presented in Table 2.

Table 2: Mean and SD of each item in C# Programming Self-Efficacy Scale

	N	Min	Max	Mean	SD
1	116	1	7	5.03	1.520
2	116	1	7	5.37	1.436
3	116	1	7	5.00	1.358
4	116	1	7	5.67	1.394
5	116	1	7	5.63	1.386
6	116	1	7	5.51	1.423
7	116	1	7	5.53	1.405
8	116	1	7	5.25	1.426
9	116	1	7	4.77	1.517
10	116	1	7	4.58	1.481
11	116	1	7	4.21	1.541
12	116	1	7	4.66	1.486
13	116	1	7	4.83	1.476
14	116	1	7	4.65	1.470
15	116	1	7	5.36	1.404
16	116	1	7	4.88	1.510
17	116	2	7	5.81	1.244
18	116	2	7	5.74	1.346
19	116	1	7	5.50	1.460
20	116	1	7	4.95	1.382
21	116	2	7	5.16	1.265
22	116	1	7	4.72	1.336
23	116	1	7	4.76	1.436
24	116	1	7	4.78	1.555
25	116	1	7	4.46	1.546
26	116	1	7	4.38	1.748
27	116	1	7	4.47	1.639
28	116	1	7	4.79	1.639
Total	116	56	196	140.44	30.53

Mean and SD of C# programming self-efficacy scores according to type of graduated high school, department and even taken programming course of the participants are presented in Table 3.

Table 3: Mean and SD of Self-Efficacy Scores according to Type of Graduated High School, Department and even Taken Programming Course of the freshman

	N	Mean	SD
Type of High School Graduated			
Computer – Electronic	84	142.97	27.80
Other Departments	32	133.81	36.40
Department			
Computer Technologies	29	138.96	32.78
Electronic Communication Technologies	87	140.94	29.92
Taken Programming Course			
Yes	29	158.41	24.96
No	87	134.45	29.97

International Journal on New Trends in Education and Their Implications
July 2014 Volume: 5 Issue: 3 Article: 15 ISSN 1309-6249

Differences in Students' C# Programming Self Efficacy with respect to Students' Departments, Type of Graduated High School

An independent t-test was performed in order to ascertain whether or not there was a significant difference between students' departments in the degree of C# programming self-efficacy. Results revealed that there wasn't any significant difference in the scores for students from Computer Technologies (CT) Department (M=138.96. SD=32.78) and students from Electronic and Communications Technologies (ECT) Department (M=140.94. SD=29.92) conditions; t(114)=-.301. p=.764. These results suggest that department does not have any effect on C# Self-Efficacy.

In addition to this, there was not a significant difference in the scores for students who graduated from high schools' Computer or Electronic Departments (M=142.97, SD=27.80) and students who from other departments of high schools (M=133.81, SD=36.40), conditions; t(114)=-1.452. p=.149.

Correlation between C# Programming Self-Efficacy and Other Variables

The correlations between C# Programming Self-Efficacy and participants' age, type of graduated high school, computer use experience in years, frequency of computer use, and programming course experience (ever taken programming course) are presented in Table 4.

Table 4: Correlations between C# programming self-efficacy and other study variables

	Age	Computer Use Experience in Years	Frequency of Computer Use	Programming Course Experience
C# Programming Self Efficacy	.021	.342**	-.119	.341**
**. Correlation is significant at the 0.01 level (2-tailed).				

There is a significant positive correlation between students' C# programming self-efficacy, experience of computer usage in years and programming course experience (p<.001). On the other hand, there is no significant correlation between C# programming self-efficacy and the other variables (p>.001). Overall, there are positive correlations between experience of computer usage in years, programming course experience and C# programming self-efficacy. Increases in experience of computer usage in years are correlated with increases in C# programming self-efficacy and taking programming course increases C# programming self-efficacy. It is clearly inferred that age and frequency of computer use don't correlate with C# programming self-efficacy.

Regression Analysis of Relationship between C# Programming Self-Efficacy and Computer Using Background

To verify whether a combination of the experience of computer usage in years and taking programming course will significantly predict C# self-efficacy, data were subjected to regression analysis. The independent variables explained only 22.3 percent of the C# self-efficacy scores (R^2 = .223, F(2. 113) = 16.199, p < .001).

Table 5: The regression results predicting C# self-efficacy from experience of computer usage in years and programming course experience

Variable	B	Standard Error	β	t stat	p value
computing experience	8.487	2.158	.327	3.933	.000
programming course experience	22.881	5.828	.326	3.926	.000

As shown in Table 5, the regression model reveals that experience of computer usage in years and programming course experience are statistically significant contributors of C# self-efficacy. Results show that experience of computer usage in years and programming course experience have almost same impact in the prediction of C# self-efficacy.

DISCUSSION

The aim of this study is to investigate predictors of C# programming language self-efficacy among vocational college students. This study focuses on the relationship between C# programming self-efficacy beliefs and age, type of graduated high school, department, frequency of computer use, experience of computer usage in years and taking programming course.

Consistent with the previous findings, the present study on vocational college domain supports existing literature on taking programming course before significantly predict students' programming self-efficacy (Ramalingan & Weidenback 1998; Weidenback. 2005). We can say that programming course experience is crucial for C# programming self-efficacy. This indicates that programming course experience continues to affect students' self-efficacy till the end of the semester. However, even at the end of the semester writing, reading and understanding C# programs seem to be challenging for the students who have never taken any programming course before.

The results of this study support findings reported in previous research (Askar & Davenport 2009) about experience of computer use in years. Students' experience of computer use in years significantly predicts C# self-efficacy. On the contrary, Jegede (2009) found years of computing experience did not predict Java self-efficacy. In this study, it is found that frequency of computer use doesn't affect programming self-efficacy consisted with Jegede (2009) and contrary to Askar and Davenport (2009).

Opposite to previous research findings (Askar & Davenport. 2009), this study shows that vocational college students' departments do not cause any difference on their C# programming self-efficacy. Also, students' age, and type of graduated high school do not significantly predict C# programming self-efficacy.

It is clearly inferred that students' department, age, type of graduated high school and frequency of computer use are not critical factors to predict students' programming self-efficacy. Most important factors to predict programming self-efficacy are programing course experience and experience of computer use in years. Understanding students' self-efficacy beliefs about computers programming is useful to design effective programming courses.

LIMITATIONS

Some difficulties have been experienced because C# programming was new for most of sophomores. To avoid such difficulties, students should be trained about programming at the first year of the school. This study was completed in a male dominated technical vocational school. Future research can take into account the effect of gender. The study, of college level students, concentrated on the predictors that determine a participant's C# programming self-efficacy beliefs rather than their effects on academic achievement. It would also be interesting to relate C# programming self-efficacy to students' academic achievement.

It is becoming crucial to gain a better understanding of student self-efficacy of C# programming and its relationships in order to improve teaching programming methods because self-efficacy is a critical factor of academic achievement. More importantly, it is essential to continuously investigate the factors such as level of education (undergraduate. postsecondary. secondary etc.), type of education (vocational. technical. engineering etc.), environment of education and family influences predicting self-efficacy.

BIODATA AND CONTACT ADDRESSES OF AUTHORS

Harun ÇİĞDEM is currently a lecturer at Computer Technologies Department, at Noncommissioned Officer College in Balıkesir-Turkey. He got PhD degree in Computer Education and Instructional Technologies in Anadolu University-Turkey in 2012. He has played a major role in the development of Course Portal of Non-Commisioned Officer College. He teaches operating systems, computer networks, and system and network administration courses in the college. His research interests involve Web Technologies, blended learning, distance education and e-learning.

Dr. Harun CIGDEM
First Lieutenant at Noncommissioned Officer School
Balıkesir- TURKEY
E. Mail: hcigdem@gmail.com

Osman Gazi YILDIRIM is currently a lecturer at Computer Technologies Department, at Noncommissioned Officer School. He got BS degree in Computer Education and Instructional Technologies in Ege University-Turkey. He teaches visual programming, object oriented programming and web programming courses in the college. His research interests involve Web Technologies, mobile learning, distance education and e-learning.

Osman Gazi YILDIRIM
First Lieutenant at Noncommissioned Officer School
Balıkesir- TURKEY
Email: ogyildirim32@gmail.com

REFERENCES

Aşkar, P. & Umay, A. (2001). Perceived computer self-efficacy of the students in the elementary mathematics teaching programme. *Journal of Hacettepe University Education Faculty 21*, 1-8.

Askar, P. & Davenport, D (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *Turkish Online Journal of Education Technology 8* (1).

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review. 84*(2). 191–215.

Bandura, A., Adams, N. E. & Beyer, J. (1977). Cognitive processes mediating behavioral change. *Journal of Personality and Social Psychology, 3,* 125-139.

Bandura, A. (1986). *Social foundations of thought and action; a social cognitive theory.* Prentice Hall. Eaglewood Cliffs. NJ.

Jegede, P.O. (2009). Predictors of Java programming self–efficacy among engineering students in a Nigerian university. *International Journal of Computer Science and Information Security, 4 (1&2).*

Ramalingam, V., La Belle, D. & Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. *ACM SIGCSE Bulletin, 36* (3), 171-175.

Ramalingam, V. & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analysis of novice programmer self-efficacy. *Journal of Educational Computing Research, 19* (4), 367-386.

Weiedenback, S. (2005). Factors affecting the success of non-majors in learning to program. *Proceedings of the first International Workshop on Computing Education Research, Seattle,* 13-24.

Zimmerman, B.J. (1995). *Self-Efficacy and educational development in self-efficacy in changing societies*. A. Bandura (ed.) Cambridge University Press. Cambridge. pp. 203-231.

Zimmerman, B. J. (2000). Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology. 25.* 82–91.

Appendix 1: C# Programlama Özyeterlik Ölçeği

Aşağıda yer alan C# işlemlerini yaparken kendinize ne kadar güvendiğinizi belirtiniz. 1 (kendime hiç güvenmem) - 7 (kesinlikle kendime güvenirim).

Kendime hiç güvenmem 1	Kendime neredeyse hiç güvenmem 2	Kendime çok az güvenirim 3	50/50 4	Kendime biraz güvenirim 5	Genellikle kendime güvenirim 6	Kesinlikle kendime güvenirim 7

1. C# kodlarını düzgün bir şekilde yazabilirim.
2. C# programlama dilinin genel yapısını ve özel kelimelerini (if, for, int vb.) anlayabilirim.
3. Mantıksal açıdan düzgün çalışan C# uygulamaları yazabilirim.
4. "Merhaba Dünya" mesajı veren bir C# uygulaması geliştirebilirim.
5. Üç sayının ortalamasını bulan bir C# uygulaması geliştirebilirim.
6. Girilen sayıların ortalamasını bulan bir C# uygulaması geliştirebilirim.
7. C#'ın mevcut metotlarını (Convert.ToDouble(), Listbox.Add() vb.) kullanabilirim.
8. Kendi C# uygulamalarımı geliştirebilirim.
9. Yabancısı olmadığım bir problemi çözebilmek için küçük çaplı bir C# uygulaması geliştirebilirim.
10. Daha önce karşılaşmadığım bir problemi çözebilmek için orta büyüklükte bir C# uygulaması geliştirebilirim.
11. Problemin detaylı bir şekilde tanımlanması durumunda, uzun ve karmaşık bir C# uygulaması geliştirebilirim.
12. C# uygulamasını, problemi parçalara ayırarak oluştururum.
13. Oluşturduğum C# uygulamasının hatalarını giderip çalışır hale getirebilirim.
14. Birkaç dosyadan (formdan) oluşan karmaşık ve uzun bir C# uygulamasının çalışma mantığını anlayabilirim.
15. Problemin çözüm yolunu bir kişi bana gösterdikten sonra bir C# uygulamasını tamamlayabilirim.
16. Kaynak olarak sadece bir C# kitabım olursa bir C# projesini tamamlayabilirim.
17. Takıldığım yerde bana yardım edecek birisi olursa bir C# projesini tamamlayabilirim.
18. Birisi bana projeye başlama konusunda yardımcı olursa bir C# projesini tamamlayabilirim.
19. C# projesini tamamlamak için çok fazla sürem olursa projeyi tamamlayabilirim.
20. Elimde sadece C#'ın kendi yardım kaynağı olması durumunda projeyi tamamlayabilirim.
21. Bir programlama projesinin herhangi bir yerinde takılırsam, bunun üstesinden gelebilmek için yollar bulabilirim.
22. Verilen bir programlama probleminin kısa sürede uygun bir çözüm yolu üretebilirim.
23. Belirli bir bitirilme süresi olan bir C# projesinde zamanımı etkin kullanabilirim.
24. Uzun ve karmaşık bir C# projesinin çalışmasını zihnimde canlandırabilirim.
25. Uzun ve karmaşık C# kodlarını daha anlaşılır bir şekilde tekrar yazabilirim.
26. Etrafımda çalışmamı engelleyen (gürültü vb.) çok fazla şey olsa bile üzerinde çalıştığım C# uygulamasına yoğunlaşabilirim.

27. Programın problem alanı (oyun programlama, internet programcılığı, sistem programcılığı, ağ programcılığı, veritabanı programlama vb.) ilgimi çekmese bile kendimi programlama yapmaya kanalize edebilirim.
28. Bir başkasının daha sonra anlayabileceği ve üzerine eklemeler yapabileceği bir C# uygulaması geliştirebilirim.