AN INVESTIGATION OF FACTORS RELATED TO SELF-EFFICACY FOR JAVA PROGRAMMING AMONG COMPUTER SCIENCE EDUCATION STUDENTS

Desmond Wesley Govender*, Sujit Kumar Basak**

Abstract

Students usually perceived computer programming courses as one of the most difficult courses since learning to program is perceived as a difficult task. Quite often students' negative perceptions on computer programming results in poor results and high drop-out rates. The purpose of this study is to examine the impact of factors that affect computer science education students' Java programming self-efficacy and the relationship between Java programming self-efficacy and students' age and gender. A questionnaire was used to gather data. A scale with thirty-two items assessing Java programming self-efficacy was adapted from Askar and Davenport's (2009) computer programming self-efficacy scale. A total of twenty students from a Computer Science Education Discipline participated in this study. Collected data were analysed using SPSS version 22.0. Descriptive statistics, reliability test, mean, standard deviation, and rotated component matrix were utilized to analyze the resulting data. Results indicated that there is not much difference between males (45%) and females (55%) Java programming self-efficacy. Furthermore, the results also indicated that programming skills and Java constructs have higher influence on the self-efficacy for Java programming among computer science education students followed by non-complexity, time consciousness, ability to recode for better understanding and self-motivation.

Keywords: Java Programming, Self-efficacy, Computer Science Education Students

*Discipline of Computer Science Education, University of KwaZulu-Natal, Pinetown, Durban, South Africa
**Faculty of Educational Sciences, University of Quebec at Montreal, Montréal, Quebec, Canada

1 Introduction

Self-efficacy is "people's judgments of their capabilities to organize and execute courses of action required in attaining designated types of performance" (Bandura, 1986: 391). Self-efficacy +plays a crucial role in the psychological construct and therefore requires attention from research as it influences "(i) the choice of activities that an individual takes part in; (ii) the amount of effort they will expend in performing a task and (iii) how long they will persevere in the face of stressful situations in completing that task (Bandura, 1977). Bandura studies (1977 and 1986) indicated that individual self-efficacy is an important factor for the performance over a wide range of situations. Askar and Davenport (2009), Aşkar and Umay (2001), Bandura, Adams, and Beyer (1977), Wiedenbeck (2005) indicated self-efficacy beliefs of individuals perceive themselves as capable of performing certain tasks or activities. Furthermore, these researchers indicated that individuals with low self-efficacy beliefs, perceived themselves less capable and less likely to attempt certain tasks or activities. According to Zimmerman (2000), educational researchers have accepted that improving students' self-beliefs about their academic capabilities does play an important role to improve their academic performance.

According to Askar and Davenport (2009), "Self-efficacy is especially important, and potentially useful, when the context relates to education. This is because the theory recognizes an individual's actual performance being influenced by their self-efficacy, hence this can affect any further performance. The theory suggests that individuals based their self-efficacy beliefs on four sources of information: i) personal experience of the skills, ii) vicarious experience-seeing people similar to oneself, perform the skills, iii) verbal persuasion, and iv) somatic and emotional states - fear, stress, also fatigue, aches and pains, etc."

According to Gist and Mitchell (1992), the main concept of the malleability is very important for the self-efficacy theory since it poses the potential to improve people's performance via increased self-efficacy. Several researchers indicated that efficacy beliefs have shown to be malleable, in the case of early stages of skill development (Bandura, 1994; Bandura, 1997; Bandura, 1986; Torkzadeh and Koufteros, 1994). "Too much malleability would not be desirable, if it led to steep drops in self-efficacy based on a single poor outcome. While intellectual ability and domain knowledge are major factors in achievement in educational settings, perceived self-efficacy also plays a strong role. Those with the same level of cognitive skill development vary in their intellectual performance depending on the strength of their self-efficacy beliefs" (Zimmerman, 1995).

2 Problem statement

According to Fang (2012), Garner (2009) and Nilsen and Larsen (2011), students experience programming difficulties with the content and algorithmic structure of the programming because of low self-efficacy and motivation. Garner (2009) indicated that learning computer programming bears a considerable cognitive load on students. Several researchers (Korkmaz, 2013; Nilsen and Larsen, 2011; Caspersen and Kolling 2009; Bergin and Reilly 2005; Dreyfus and Dreyfus, 1986) indicated that students usually perceived computer programming courses as one of the most difficult courses as learning to program is perceived as a difficult task, compared to the other subjects, high drop-out and the failure rates are very common. In this relation, learning computer programming requires the process of acquiring skills which are considered difficult and requires a great amount of effort. According to Gomes and Mendes (2007), many problems are related to the acquiring of computer programming skills for academic success and the level of student's satisfaction. Studies by Tan et al. (2009) stated that during the learning phase of computer programming, difficulties faced by students contributes directly to the development of the negative perception on computers as a discipline. This phenomenon is usually caused by a sense of misjudgement among students, that learning and acquiring a competence in programming is a highly difficult process. However, as a consequence, students unconsciously reject the process of learning programming. Furthermore, their studies also indicated it is a crucial task to determine beforehand, students' perception of programming and take countermeasures to tackle and address the problems associated with these perceptions.

3 Research question

What factors affect computer science education students Java programming self-efficacy?

4 Aim and objectives

The aim of this study is to evaluate the impact of factors that affect computer science education students' Java programming self-efficacy. This aim is achieved through the following specific objectives:

- To identify the self-efficacy factors influencing computer science education students Java programming;
- To analyse the impact of self-efficacy factors on computer science education students Java programming;
- To examine the activities and strategies that universities should take in order to improve the levels of satisfaction among computer science education students of their Java programming self-efficacy.

5 Literature review

A study was conducted by Korkmaz and Altun (2014) on engineering student's C++ computer programming self-efficacy levels and their sample size of the study consisted of 378 engineering students. The results of the study indicated that the scale is reliable and valid, and it can be used for the measurement of self-efficacy of engineering students in a Turkish cultural environment. Furthermore, their study also indicated that self-efficacy perception of students in computer engineering is found to be higher than that of the students in electrical-electronic engineering. Ramalingam and Wiedenbeck (1998) conducted a study and their research results indicated that differences in self-efficacy between the pre-test and the post-test was usually found among students with low self-efficacy. Moreover, their studies also indicated that no substantial difference was found between males and females.

Studies conducted by Anastadiadou and Karakos (2011), Erdogan et al., (2008), Sacks et al., (1993), and Austin (1987) indicated that several factors affects the success for learning , but it is generally accepted that attitude and self-efficacy perception are the most important factors among others. Research results from Levine and Donitsa-Schmith (1998) indicated that computer competence and computer literacy are not just related to level of knowledge, constraints, applications, and effect of the computer but, at the same time, it is directly related to the individual's attitudes towards the computer.

According to Askar and Davenport (2009), self-efficacy directly affects the process of acquiring new skill and using a new skill. They also stated that self-efficacy can be used as a tool and a reliable one in order to predict a person's performance. Furthermore, their studies indicated that self-efficacy of the students were influenced by their computer experience and their computer skills. Moreover, their studies stated that students' gender and the family usage of computers did not affect students' self-efficacy (Askar and Davenport, 2009). Studies by Ramalingam, La Belle, and Wiedenbeck (2004) stated that previous programming experience has an influence on programming self-efficacy.

A study was conducted by Jegede (2009) in a university in Nigeria on engineering students' Java programming self-efficacy with students' programming experience. The research results indicated that there is a relationship between the students' Java programming self-efficacy and each of the computer use and programming experience factors. Wiedenbeck (2005) presented a view on the self-efficacy as a factor of success when comparing students at the same level of the cognitive skills development, a student with higher self-efficacy beliefs is more likely to take on more advanced progressive challenges and finally will strive harder to reach the goals. Furthermore, Wiedenbeck (2005) and Askar and Davenport (2009) also indicated that with the low self-efficacy beliefs an individual tends to exaggerate problems and undergo stress and depression that can make a solvable problem impossible. According to Cassidy and Eachus (2002), higher levels of perceived self-efficacy correlate to generate the motivational efforts and the perseverance. Askar and Davenport (2009) stated that self-efficacy has emerged as an important means of understanding the predicting of a person's performance. On the other hand, according to Jegede (2007), higher levels of computer self-efficacy correspond to greater achievement of computer competence.

According to Korkmaz (2011), a person's self-efficacy perception is the self-evaluation of his or her competence to conduct the task successfully. Furthermore, it is considered a prediction of a person's aptitude for what he or she can accomplish, in terms of his or her competence in order to complete a particular task. Moreover, a person can have all the necessary qualities to complete a task, but the deficiency in self-belief and lack of motivation can raise a failure. Further, Asker and Davenport (2009) indicated that self-efficacy has the direct effect to process a new acquiring skill. In this connection, the level of the self-efficacy can be used as a tool and a reliable indicator in terms of a person's performance.

6 Methodology

The underlying research design is based on the use of validated questionnaires from Ashkar and Davenport (2009). Collected data were analyzed using SPSS version 22.0. The next section discusses these questionnaires and their measurements.

6.1 Sample and procedure

This study was experimental and conducted within a public university in KwaZulu-Natal, South Africa. The sample consisted of 20 computer science education students. Each participant's participation was purely voluntary and they were assured of anonymity. The questionnaire, which is described in more detail in the following section, was administered to participants in the first week to the second week of May 2015.

6.2 Questionnaire

The questionnaire was developed to capture information relevant to the study and consisted of two parts. Part One sought information on factors affecting, Part Two consisted of questions regarding demographics. Each participant was allocated 20 minutes to fill the questionnaire. There was a total of 34 questions in the survey. In 31 of the questions, the students were asked to answer (on a scale 1 to 7) how well they think they meet five specific course goals of the introductory programming course:

a) Analyze and design solutions for simple problems

b) Formulate a strategy for managing the larger problems

c) Systematic debugging

d) Read, understand and modify the small parts of large amounts of code written by others

e) Explain general principles for how a computer is structured and operates

Three of the questions asked where for gender, programming experience and what student group they belonged to.

7 Results

7.1 Demographics

Demographics consist of three items, namely, age, gender, and year of study. All the three items have been explained below with tables and graphs.

7.1.1 Age

Table 1 shows that all the respondents are between 20 to 30 years old. A graphical representation of this distribution is depicted in Figure 1.

Table 1. Age frequency distribution

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	20-30 years	20	100.0	100.0	100.0

7.1.2 Gender

Table 2 and Figure 2 represents that more than half of the respondents (55%) are female. On the other hand, 45% of the respondents are male. A graphical representation of this distribution is depicted in Figure 2.

Figure 1. Age distribution

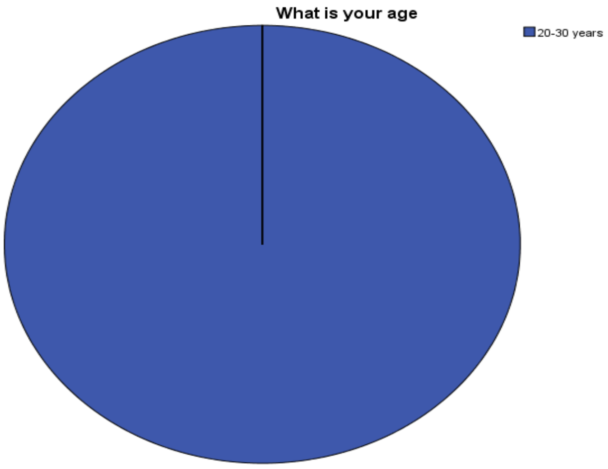
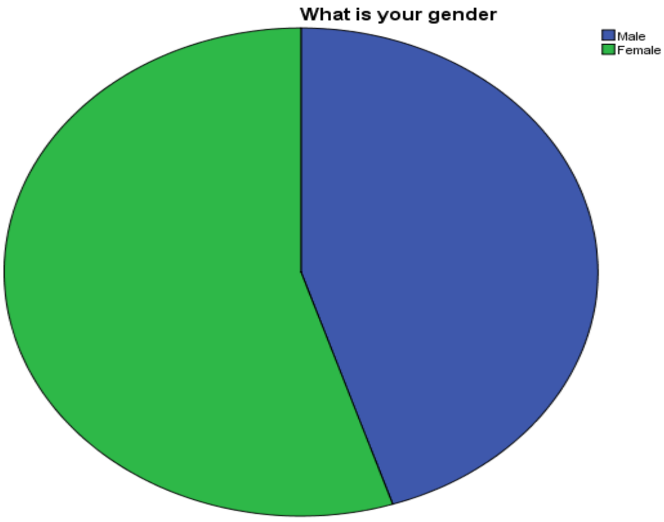


Table 2. Gender frequency distribution

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	Male	9	45.0	45.0	45.0
	Female	11	55.0	55.0	100.0
	Total	20	100.0	100.0	

Figure 2. Gender distribution



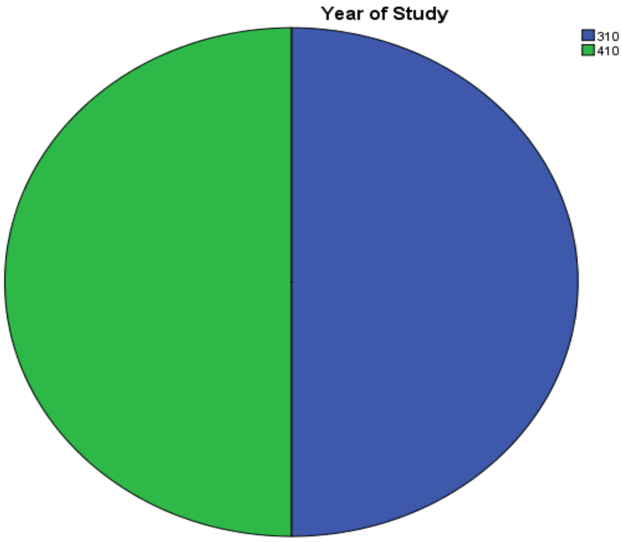
7.1.3 Year of study

Table 3 and Figure 3 represents that half of the respondents were from the level of study 310 and half of the respondents were from the level of study 410. A graphical representation of this distribution is depicted in Figure 3.

Table 3. Year of study frequency distribution

		Frequency	Percent	Valid Percent	Cumulative Percent
Valid	310	10	50.0	50.0	50.0
	410	10	50.0	50.0	100.0
	Total	20	100.0	100.0	

Figure 3. Year of study distribution



8 Reliability

Reliability testing of the questionnaire data was done by calculating Cronbach's coefficient alpha (α) for each Likert scale based section of the questionnaire. Data from the questionnaire was reliable because the Table 4 shows that the Cronbach's Alphas is 0.973 for the 32 items.

Table 4. Cronbach's Alpha

Cronbach's Alpha	Cronbach's Alpha Based on Standardized Items	N of Items
.973	.973	32

9 Mean and standard deviation of self-efficacy

Mean and standard deviations of the self-efficacy scores of all the 32 items are given in Table 5. There was a significant difference between each item of the questions of the self-efficacy for Java programming.

10 Rotated component matrix

The researchers also performed the testing of the overall validity of the questionnaire data by factor analyzing all the reliable and valid items from the Likert scale based on variables and the results was as follows:

- Number of items 32
- Number of components extracted is 6

Table 6 shows the rotated matrix of the reliable and valid items from the Likert scale based variables. The extraction of six components of this factor analysis confirms that the Likert scale based variables of the questionnaire are indeed valid to represent all the variables. Six components are namely programming skill (component 1), Java constructs (component 2), non-complexity (component 3), time consciousness (component 4), ability to recode for better understanding (component 5) and self-motivation (component 6).

In the case of programming skill (component 1), Table 6 shows that out of 32 items only four items, namely Q9 (.754), Q10 (.877), Q12 (.713), and Q27 (.777) have significantly influenced students in the computer science education discipline in terms of self-efficacy for Java programming. *In the case of Java constructs (component 2),* Table 6 shows that out of 32 items only four items, namely Q2 (.800), Q7 (.866), Q14 (.878), and Q15 (.890) have significantly influenced students in the computer science education discipline in terms of self-efficacy for Java programming. *For the non-complexity (component 3),* Table 6 shows that out of 32 items only three items, namely Q4 (.912), Q5 (.883), and Q6 (.793) have significantly influenced students in the computer science education discipline in terms of self-efficacy for Java programming. *For the time consciousness (component 4),* Table 6 shows that out of 32 items, only two items, namely Q23 (.855) and Q24 (.818) have significantly influenced students in the computer science education discipline in terms of self-efficacy for Java programming. *In the case of time consciousness (component 5),* Table 6 shows that out of 32 items only one item, namely Q30 (.760) have significantly influenced students in the computer science education discipline in terms of self-efficacy for Java programming. *Finally, in the case of Self-motivation (component 3),* Table 6 shows that out of 32 items only one item, namely Q31 (.908) have significantly influenced students in the computer science education discipline in terms of self-efficacy for Java programming. Among all six components only first two components have the highest influence on the self-efficacy for the Java programming followed by component 3, component 4, component 5 and component 6.

Table 5. Mean and standard deviation of self-efficacy according to questionnaire items

Item Statistics			
	Mean	Std. Deviation	N
Q1	5.50	.946	20
Q2	5.25	1.293	20
Q3	5.30	1.081	20
Q4	6.40	.754	20
Q5	6.00	1.451	20
Q6	5.70	1.418	20
Q7	5.05	1.317	20
Q8	4.45	1.761	20
Q9	5.85	1.348	20
Q10	5.05	1.538	20
Q11	5.10	1.651	20
Q12	4.80	1.704	20
Q13	4.75	1.333	20
Q14	5.60	1.536	20
Q15	5.40	1.392	20
Q16	5.75	1.020	20
Q17	5.20	1.005	20
Q18	4.50	1.357	20
Q19	5.40	1.095	20
Q20	5.15	1.309	20
Q21	5.55	1.504	20
Q22	5.55	1.050	20
Q23	5.55	1.099	20
Q24	5.40	1.095	20
Q25	5.00	1.214	20
Q26	4.50	1.318	20
Q27	4.80	1.473	20
Q28	4.40	1.046	20
Q29	4.00	1.376	20
Q30	4.55	1.317	20
Q31	4.90	1.483	20
Q32	5.00	1.170	20

Table 6. Rotated matrix for the six Likert scale based variables

Rotated Component Matrix						
	Component					
	1	2	3	4	5	6
Q1	.474	.589	.495			
Q2		.800				
Q3	.510	.461	.470			
Q4			.912			
Q5			.883			
Q6			.793			
Q7		.866				
Q8		.608				
Q9	.754					
Q10	.877					
Q11	.672					
Q12	.713	.469				
Q13	.508	.612				
Q14		.878				
Q15		.890				
Q16	.490					
Q17	.688					
Q18	.638	.474				
Q19	.571					
Q20	.472	.548		.466		
Q21						.587
Q22					.613	
Q23				.855		
Q24				.818		
Q25	.682			.520		
Q26	.534					.520
Q27	.777					
Q28	.699					
Q29					.689	
Q30					.760	
Q31						.908
Q32	.518				.493	

11 Conclusion and discussion

The aim of this study was to investigate predictors of Java programming language, self-efficacy among computer science education students. This study focused on the relationship between Java programming, self-efficacy beliefs, as well as age and gender. Our results indicated that programming skills and Java constructs have high influence for Java programming in computer science education students followed by non-complexity and time consciousness. Students were less confident on the ability to recode for better understanding and self-motivation since only one item contributed to the factor.

Overall, our initial results confirm that the relevance of self-efficacy to the acquisition of programming skills, Java constructs, non-complexity, time consciousness, ability to recode for better understanding, and self-motivating is in line with Bandura's theory. However, consistent with the previous findings, the present study of computer science education students supports existing literature on taking a programming course before significantly predicting students' programming self-efficacy (Ramalingam and Wdidenback, 1998; Weidenback, 2005; Askar and Davenport 2009).

On the contrary, Jegede (2009) found years of computing experience did not predict Java self-efficacy. However, we can say that the programming course experience continues to affect students' self-efficacy until the end of the year. Furthermore, even at the end of the semester, reading and understanding Java programming seems to be challenging for the students who have never taken any programming course before.

12 Limitations

Although this study is limited by the time limits under which it was carried out, the preliminary results show that there is a strong influence with programming skills, Java constructs, non-complexity, time consciousness, ability to recode for better understanding, and self-motivation for self-efficacy of Java programming in computer science education students. For future studies, it would be very interesting to carry out the longitudinal study where one can follow the same group of students rather than asking students from two different years with regard to their self-efficacy beliefs. Some difficulties have been experienced because Java programming was new for most of the students in the 310 level. The study, of university level students, concentrated on the predictors that determine a participant's Java programming self-efficacy beliefs rather than their effects on academic achievement. It would also be interesting to relate Java programming self-efficacy to students' actual academic achievement.

References

1. Anastadiadou, S. D. and Karakos, A. S. (2011), "The beliefs of electrical and computer engineering students' regarding computer programming", The International Journal of Technology, Knowledge and Society, Vol. 7, No. 1, pp. 37-51.
2. Aşkar, P. and Umay, A. (2001), "Perceived computer self-efficacy of the students in the elementary mathematics teaching programme", Journal of Hacettepe University Education Faculty 21, pp. 1-8.
3. Askar, P. and Davenport, D. (2009), "An investigation of factors related to self-efficacy for java Programming among engineering students", The Turkish Online Journal of Educational Technology, Vol. 8, No. 1, pp. 26-32.
4. Austin, H. S. (1987), "Predictors of pascal programming achievement for community college students", Proceedings of the eighteenth SIGCSE technical symposium on computer science education, Missouri, United States, pp. 161-164.
5. Bandura, A. (1977), "Self-Efficacy: Toward a unifying theory of behavioural change", Psychological review, Vol. 84, No. 2, pp. 191-215.
6. Bandura, A. (1986), Social Foundations of Thought and Action, Prentice hall, Englewood Cliffs, NJ.
7. Bandura, A. (1994), Self-efficacy. In V.S. Ramachaudran (Ed.), Encyclopedia of Human Behavior, Vol. 4, Academic Press, NY, 71-81.
8. Bandura, A. (1997), Self-efficacy: The exercise of control. W. H Freeman, NY.
9. Bandura, A. Adams, N. E. and Beyer, J. (1977), "Cognitive processes mediating behavioral change", Journal of Personality and Social Psychology, Vol. 35, No. 3, pp. 125-139.
10. Bergin, S., and Reilly, R. (2005), The Influence of Motivation and Comfort-level on Learning to Program, In University of Sussex, Brighton, UK.
11. Caspersen, M. E., and Kolling M. (2009), STREAM: A First Programming Process. ACT Transaction on Computing Education, 9, 1-29.
12. Cassidy, S., and Eachus, P. (2002), "Developing the computer User Self-Efficacy (CSUE) Scale: Investigating the Relationship between Computer Self-Efficacy, Gender and Experience with Computers", Educational Computing Research, Vol. 26, No. 2, pp. 133-153.
13. Dreyfus, H., and Dreyfus, S. (1986), Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer, New York: Free Press.
14. Erdogan, Y., Aydin, E., and Kabaca, Y. T. (2008), "Exploring the psychological predictors of programming achievement", Journal of instructional psychology, Vol. 35, No. 3, pp. 264-270.
15. Fang, X. (2012), "Application of the participatory method to the computer fundamentals course, Affective Computing and Intelligent Interaction", Advances in Intelligent and Soft Computing, 137, pp. 185-189.
16. Garner, S. (2009), Learning to Program from Scratch, 9th IEEE Int. Conference on Advanced Learning Technologies, 451-452, Doi: 10.1109/ICALT.2009.50.
17. Gist, M. E., and Mitchell, T. R. (1992), "Self-efficacy: A Theoretical Analysis of Its Determinants and Malleability", Academy of management review, Vol. 17, No. pp. 183-211.
18. Gomes, A. and Mendes, A. J. (2007). "Learning to program-difficulties and solutions. International conference on engineering education, ICEE. 3-7 Temmuz, Coimbra, Portugal. Available at http://ineer.org/Events/ICEE2007/papers/411.pdf
19. Jegede P.O. (2007), "Factors in computer Self-Efficacy among South-Western Nigeria College of Education", Teachers Journal of Psychology in Africa, Vol. 17, No. 1.
20. 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).
21. Korkmaz, Ö. (2011), "Study of validity and reliability of self-efficacy scale of teaching material utilization", Educational research review (ERR). Vol. 6, No. 15, pp. 843-853. Doi: 10.5897/ERR11.174
22. Korkmaz, Ö. (2013), "Students' difficulties in an opinions about designing algorithms according to different instructional applications", Energy education science and technology Part B: Social and Educational Studies, Vol. 5, No. 1, pp. 209-218.
23. Korkmaz, Ö. and Altun, H. (2014), "Adapting computer programming self-efficacy scale and engineering students' self-efficacy perceptions", Participatory Educational Research (PER), Vol. 1, No. 1, pp. 20-31.
24. Levine, T., and Donitsa-Schmidt, S. (1998), "Computer use, confidence, attitudes, and knowledge: a causal analysis", Computers in Human Behavior, Vol. 14, No. 1, pp. 125-146.
25. Nilsen H. and Larsen A. (2011), "Using the personalized system of instruction in an introductory programming course", NOKOBIT, 27-38. November 21-23.
26. Ramalingam, V. and 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, Vol. 19, No. 4, pp. 367-386.
27. Sacks, C., Bellisimo, Y., and Mergendoller, J., (1993), "Attitudes toward computers and computer use: the issue of gender", Journal of Research on Computing in Education, Vol. 26, No. 2, pp. 257-269.

28. Tan, P.H. Ting C.Y. and Ling, S.W. (2009), "Learning difficulties in programming courses: undergraduates' perspective and perception", International Conference on Computer Technology and Development, pp 42-46. DOI: 10.1109/ICCTD.2009.188
29. Torkzadeh, G., and Koufteros, X., (1994), "Factorial validity of a computer self-efficacy scale and the impact of computer training", Educational and psychological measurement, Vol. 54, No. 3, pp. 813-821.
30. Wiedenbeck, S. (2005), "Factors affecting the success of non-majors in learning to program", Proceedings of the first International Workshop on Computing Education Research, Seattle, pp. 13-24.
31. Zimmerman, B. J. (2000), "Self-efficacy: An essential motive to learn", Contemporary Educational Psychology, Vol. 25, No. pp. 82–91.
32. 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.