AN INVESTIGATION OF FACTORS RELATED TO SELF-EFFICACY FOR JAVA PROGRAMMING AMONG ENGINEERING STUDENTS

Petek ASKAR
Department of Computer Education and Instructional Technology, Faculty of Education, Hacettepe University, Beytepe, 06530,Ankara, Turkey
paskar@hacettepe.edu.tr

David DAVENPORT
Department of Computer Engineering, Faculty of Engineering, Bilkent University, 06530, Ankara, Turkey
david@bilkent.edu.tr

ABSTRACT
The purpose of this study was to examine the factors related to self-efficacy for Java programming among first year engineering students. An instrument assessing Java programming self-efficacy was developed from the computer programming self-efficacy scale of Ramalingam & Wiedenbeck. The instrument was administered at the beginning of the course via Internet with a questionnaire concerning gender, department, computer skills, computer experience, frequency of computer use and family background. Results indicated that self-efficacy of males were stonger than females, 11. 8 % of the variance in self-efficacy was explained by computer experience, the correlation coefficient calculated with the regression factor score of computer skills and self-efficacy scores was statistically significant. In addition siblings use of computers was significant and the mother's role was critical.

KEYWORDS: Gender studies, programming , higher education, engineering students, self-efficacy

1. INTRODUCTION
Self-efficacy theory has emerged as an important means of understanding and predicting a person's performance. Bandura defined perceived self-efficacy as "...people's judgements of their capabilities to organize and execute courses of action required to attain designated types of performances" (Bandura, 1986, p.391). Self-efficacy mediates between an individual's knowledge and their actions. "Knowledge, skill, and prior attainments are often poor predictors of subsequent attainments because the beliefs that individuals hold about their abilities and about the outcome of their efforts powerfully influence the ways they will behave" (Pajares, 1996). In other words, one may possess the necessary knowledge or skills needed to perform a particular task, but still not succeed owing to self-doubt, lack of motivation, or other contextual factors. Self-efficacy, then, is a measure of an individual's belief in their ability to act successfully in a particular context. As such, it exercises a powerful influence over what an individual chooses to do and the courses of action they pursue.

Self-efficacy is especially important, and potentially useful, when the context relates to education. This is because the theory recognises also that an individual's actual performance influences their self-efficacy, and hence can affect any future performances. Thus learners with high self-efficacy are more likely to undertake challenging tasks and to expend considerably greater effort to complete them in the face of unexpected difficulties, than those with lower self-efficacy. Moreover, the learning achievements that result, particularly in the face of adversity, lead naturally to improvements in the individual's self-efficacy, and so, potentially, to further successes in the future. In contrast, those with low self-efficacy tend to view tasks as being more difficult than they really are and consequently may experience stress, depression and adopt such a narrow vision of a problem as to be unable to solve it. They are thus less likely to persevere or to seek out new opportunities for learning.

Theory suggests that individuals base their self-efficacy beliefs on four sources of information, 1) personal experience of the skill, 2) vicarious experience –seeing people similar to oneself perform the skill, 3) verbal persuasion, and 4) somatic and emotional states –fear, stress, also fatigue, aches & pains, etc. Research by Bandura (1986) showed that efficacy perceptions develop gradually with the attainment of skills and experience.. Individuals form their self-efficacy beliefs by interpreting information primarily from their previous experience (Bandura, 1994, 1995) . In addition to mastery experience, self-efficacy appraisals are partly influenced by vicarious experience of observing others perform similar tasks.

Most researchers have explored the relationship among academic performance, self-efficacy and demographical variables on different fields. Computer self-efficacy is one of them which has attracted several researchers from a variety of disciplines (Albion, 2001; Askar and Umay, 2001;Delcourt and Kinzie, 1993; Karsten and Roth, 1998, Compeau and Higgins, 1995; Kurbanoglu, 2003, Usluel, 2007).

Despite this, surprisingly little attention has been paid to its role in the acquisition of computer programming skills. Ramalingam & Wiedenbeck (1998) developed and validated a self-efficacy scale for the C++ language programming. Their results using this scale seem to support the applicability of self-efficacy to C++ programming. They found that the self-efficacy of males and females did not differ substantially in practical terms. Female self-efficacy scores were found to be significantly lower than those of males, even after 12 weeks of instruction, however, the difference was small accounting for only 2% of the variance. They explained this by noting that the female students in their study were self-selected and probably had better maths skills and more computer experience than average. Since such abilities are assumed to be highly relevant to the acquisition of computer programming skills, the lack of practical difference was perhaps not surprising. Of more importance was their observation of a general increase in self-efficacy on the post-instruction measure, independent of gender, but with the greatest gains seen in those with lower initial self-efficacy scores. This is in line with theory, which suggests that the self-efficacy of beginning students is highly responsive to performance accomplishments in the early stages of skill acquisition.

In another study (Ramalingam, V., La Belle, D. and Wiedenbeck, S 2004), the researchers investigated the effects of self-efficacy and mental models of programming. The results showed that self-efficacy for programming influenced by previous programming experience.
This study aims to understand the factors related to some aspects of personal experience of the skill and vicarious experience which account for the variability in the perceived self-efficacy within the context of computer programming.

Given the huge global demand for computer literate engineers, it is vital to understand what factors influence an individual's choice of computing as a profession and subsequently affect their ability to acquire computer programming skills. Nowadays, computer programming is a common, often mandatory part, of an undergraduate engineering education. Yet, in contrast to subjects such as maths and physics, students tend to perceive programming courses as being somewhat difficult. In part, this is undoubtedly due to unfamiliarity with the subject, since computing (unlike maths and physics) has not traditionally been part of the high school curriculum. But in spite of the fact that students are increasingly likely to have had prior exposure to computers both at home and school –albeit generally as a tool for writing reports, communicating with friends or simply playing games, rather than as something that they can program to perform new tasks– such negative perceptions seem to persist. This initial study thus focuses on the relationship between programming self-efficacy beliefs and other variables (including gender, family issues, subject/department choice, prior computing skills, and frequency of computer use). The study, of university-level engineering students, concentrates on the factors that determine an individual's self-efficacy beliefs rather than their effects on performance.

2. METHOD
A group of freshmen engineering students enrolled on an introductory Java computer programming course, were asked to complete a questionaire and a self-efficacy scale at the very beginning of the course. The resulting data were used to investigate the following research questions, which form the core of the study:

1) Is there a significant difference between female and male students' self-efficacy for Java programing?
2) Are there significant differences in self-efficacy of Java programing students from the various engineering departments?
3) How do prior computing skills & frequency of computer use affect self-efficacy beliefs?
4) How does family (mother, father, & sibling) usage of computers affect self-efficacy beliefs?

2. 1 Participants
The subjects in this study were 326 engineering and science students, (200 first year students from the computer, electronics, and industrial engineering departments, and 20 first year science students, all of whom were enrolled in an introductory Java programming course, plus 106 second, third and fourth year computer engineering students who volunteered to take part.) All students completed the self-efficacy scale and returned demographic data including gender, age, department, family issues, years of computer experience, and skills at computer usage.

All the participants were students at Bilkent University, in Ankara, Turkey. As such, they should be considered very good academically, having achieved scores within the top 3 % of the national university entrance examination (taken annually by some 1.5 million students!) All instruction at Bilkent is in English, and students are required to pass a English language proficiency exam prior to entering their departments. Computer programming is a required course for all engineering students.

2.3 Instrumentation
An instrument assessing Java programming self-efficacy JPSES was developed from the computer programming self-efficacy scale of Ramalingam & Wiedenbeck (1998). The overall reliability of the self-efficacy scores for their C++ scale was 0.98. The corrected item-total correlations ranged between 0.5 and 0.84. The new scale for Java Programming consisted of 32 items and the reliability of the scores was 0.99 (taken across all 326 students). The scale, which was in English, is giveb in Appedix 1. The participants were given instructions to: "*Rate your confidence in doing the following Java programming related tasks using a scale of 1 (not all confident) to 7 (absolutely confident).*" In addition to the scale, a questionnaire for collecting information related to gender, department, prior computer experience, computer skills and family computer usage was prepared and delivered to the students via the Internet.

3. RESULTS
Analysis of variance, principal components and regression analysis were undertaken using SPSS to investigate any relationship among self-efficacy for Java programming and the other variables obtained from the questionnaire. The overall self-efficacy scores were used in the analyses.

Table 1: Mean and SD of Self-Efficacy Scores according to Gender and Deparment of the Freshman

	N	Mean	SD
Gender			
Female	42	43,64	20,24
Male	167	74,05	52,57
Deparment			
Industrial Engineering	58	52,17	33,83
Electrical-Electronics En.	67	69,76	45,97
Computer Engineering	75	74,73	58,65

3. 1 Gender Issues
Means and standard deviations of the self-efficacy scores of female and male student are given in Table 1. There was a significant difference between males and females with regard to the self-efficacy for Java programming. (t=5.929, p=0.00) with the self-efficacy of males being higher than that of females.

3. 2 Department
The main effect of department was significant. ($F_{(2;303)}=44.185$, $p<0.01$) The multiple comparison analysis showed that computer engineering students' self-efficacy scores were significantly higher than that of students from the other engineering departments.

3. 3 Computer Experience
Simple Regression analysis revealed that the number of years of experience a student had with computers had a significant linear contribution to their self-efficacy scores. (R= 0.344, $F_{(1,304)}=40.682$, $p<0.001$). Indeed 11.8 % of the variance in self-efficacy was explained by the number of years of computer experience a student had. As the computer experience increases there is a tendency to gain self-efficacy in programming.
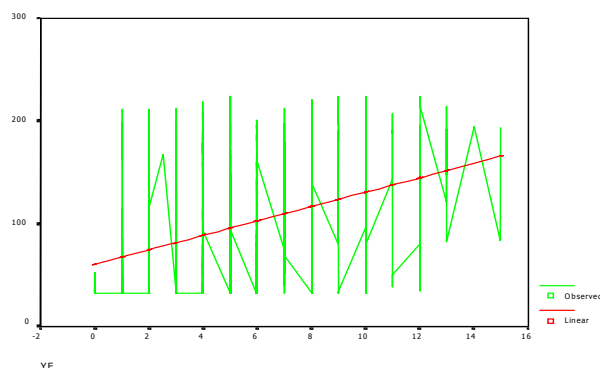

Figure 1. Linear curve fit x= years of experience, y= self-efficacy scores

3.4 Computing Skills

Computer skills were measured on chat, e-mail, word-proceesing, spreadsheet, powerpoint, web-design, databases and programming. The principal component analysis revealed only one factor with an eigen value higher than 1. The factor loadings were ranged between 0.54 (skill in chat), and 0.802 (skill in word-processing).
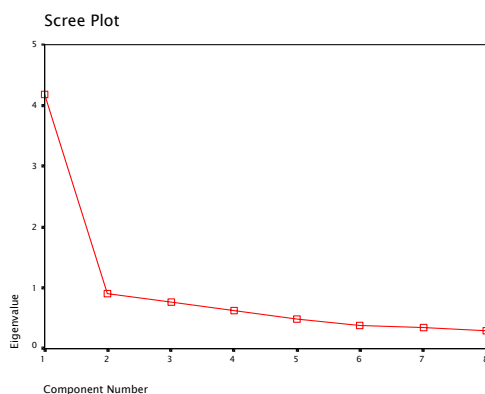

Figure 2: Scree Plot of Computer Skills

The correlation coefficient calculated with the regression factor score and self-efficacy score was statistically significant (r=0.592, p<0.01).

Table 2: Factor loadings on the first factor (Principal Component Analysis)

Skill in Computers	Factor loadings on Component 1
Chat	0,535
E-mail	0,746
Word processing	0,802
Spreadsheet	0,679
Presentation	0,781
Data-bases	0,741
Internet	0,758
Programming	0,703

3.5 Frequency of Computer Use

Significant differences were found between the students who used computers everyday and those who used them only a few times in a week (t=8,216, p<0.01; with the more frequent users having higher self-efficacy scores (means are 109,66 and 57,40 respectively).

3.6 Family Computer Usage

Table 3: Self-efficacy Scores According to Family Computer Use

	N	Mean	SD
Mother			
Using	29	86,45	60,89
Not Using	180	64,96	46,86
Father			
Using	78	67,76	50,28
Not Using	131	68,05	48,98
Sibling			
Using	139	73,29	52,96
Not Using	70	57,31	39,50

The self-efficacy scores of the students who reported that their mothers used computers were significantly higher than those students whose mothers did not use computers (t= 2.196, p=0.029). In addition, the students whose brother or sisters used computers got statistically higher self-efficacy scores than students who reported

that their siblings did not use computers (t=2.452, p=0.015). On the other hand, no statistically significant differences were observed between the self-efficacy scores of the students whose fathers used computers and those who didn't (t=0.042, p=0.967). Overall, the highest self-efficacy scores was clearly that of the mother's computer use.

4. DISCUSSION AND CONCLUSION

Since its emergence as part of Bandura's Social Cognitive Theory in the late 1970's, self-efficacy has been the subject of considerable interest amongst education researchers. Because self-efficacy is based on self-perceptions regarding particular behaviors, the construct is considered to be situation specific or domain sensitive. That is, an individual may exhibit high levels of self-efficacy within one domain while exhibiting low levels within another domain (Cassidy and Eachus, 1998).

This initial study thus focuses on the relationship between programming self-efficacy beliefs and gender, family issues, subject/department choice, prior computing skills, and frequency of computer use. The study, of university-level engineering students, concentrates on the factors that determine an individual's self-efficacy beliefs rather than their effects on performance.

Our results indicated that female students had significantly lower initial self-efficacy beliefs compared with those of their male peers. In our case all students were required to take the course, but it may be considered similar to Ramalingam & Wiedenbeck's study since all students entering the engineering departments have extremely high scores on the university entrance exam indicating very good mathematical & analytic skills. In this sense, they may even be considered gifted students and previous studies have shown that gifted girls in particular are prone to underestimate their self-efficacy (compared with the more normal overestimation of ability seen in most youngsters.)

One of the major areas of self-efficacy research in academic settings has been to explore its link with subject/career choice. Our results seem to confirm such a link, the computer engineering students recording higher self-efficacy for computer programming (their chosen major) than those who selected electronics or industrial engineering. On the other hand, the self-efficacy scores of all the students was markedly low. This is probably a result of two factors. First, as intimated previously, few students have any background in programming and are thus being forced to base their beliefs on other factors (especially their maths ability and prior computer usage.) Secondly, students in Turkey often select their major subject area on the basis of social and family pressures, departments being chosen simply on the basis of the score achieved in the university entrance exam, without any real notion of what the subject might involve.

Our study found a relatively high correlation of 0.59 between self-efficacy beliefs and prior (non-programming) experience with computers. The frequency of computer use also produced improved self-efficacy although the overall scores were still quite low. As noted above, this may be due to a number of factors, in particular, the fact that students are being forced to make judgements without direct prior experience and thus have to extrapolate from experiences they see as possibly relevant into a new unknown domain.

Perhaps the most interesting aspect of our study relates to the effects of computer use by family members on student self-efficacy scores. Whilst the father appears to have practically no influence, and siblings usage is certainly significant, the mother's role is especially valuable. We may explain this by assuming that mothers are generally viewed by their offspring as being equal or less technologically literate than they are. In this case, the mothers are effectively peers and so the result is probably in line with the theory which views only those of equal or less ability as relevant to ones beliefs. In contrast, the fathers knowledge/competence has little relation to their skill acquisition since they do not so relate.

Overall, our initial results appear to confirm the relevance of self-efficacy to the acquisition of Java programming skills and are in line with Bandura's theory. Clearly, there is a need for further detailed investigations, particularly of the family influences.

Appendix 1.
Java Programming Self-Efficacy Scale

Rate your confidence in doing the following Java programming related tasks using a scale of 1 (not at all confident) to 7 (absolutely confident). **If a specific term or task is totally unfamiliar to you, please mark 1.**

Not confident at all	Mostly not confident	Slightly confident	50/50	Fairly confident	Mostly confident	Absolutely confident
1	2	3	4	5	6	7

1. I could write syntactically correct **Java** statements. _____
2. I could understand the language structure of **Java** and the usage of the reserved words. _____
3. I could write logically correct blocks of code using **Java** _____
4. I could write a **Java** program that displays a greeting message. _____
5. I could write a **Java** program that computes the average of three numbers. _____
6. I could write a **Java** program that computes the average of any given number of numbers. _____
7. I could use built-in functions that are available in the various Java **applets**.. _____
8. I could build my own **Java applets**. _____
9. I could write a small Java program given a small problem that is familiar to me.
10. I could write a reasonably sized Java program that can solve a problem this is only vaguely familiar to me. _____
11.   I could write a long and complex Java program to solve any given problem as long as the specifications are clearly defined. _____
12.   I could organize and design my program in a modular manner. _____
13.   I could understand the object-oriented paradigm. _____
14.   I could identify the objects in the problem domain and could declare, define, and use them. _____
15.   I could make use of a pre-written function, given a clearly labeled declaration of the function. _____
16.   I could make use of a class that is already defined, given a clearly labeled declaration of the class. _____
17.   I could debug (correct all the errors) a long and complex program that I had written and make it work. _____
18.   I could comprehend a long, complex multi-file program. _____
19.   I could complete a programming project if someone showed me how to solve the problem first. _____
20.   I could complete a programming project if I had only the language reference manual for help. _____
21.   I could complete a programming project if I could call someone for help if I got stuck. _____
22.   I could complete a programming project once someone else helped me get started. _____
23.   I could complete a programming project if I had a lot of time to complete the program. _____
24.   I could complete a programming project if I had just the built-in help facility for assistance. _____
25.   While working on a programming project, if I got stuck at a point I could find ways of overcoming the problem. _____
26.   I could come up with a suitable strategy for a given programming project in a short time. _____
27.   I could manage my time efficiently if I had a pressing deadline on a programming project. _____
28.   I could mentally trace through the execution of a long, complex multi-file program given to me. _____
29.   I could rewrite lengthy and confusing portions of code to be more readable and clear. _____
30.   I could find a way to concentrate on my program, even when there were many distractors around me. _____
31.   I could find ways of motivating myself to program, even if the problem area was of no interest to me. _____
32.   I could write a program that someone else could comprehend and add features to at a later date. _____

REFERENCES
Albion P.R. (2001) "Some factors in the development of self-efficacy beliefs for computer use among teacher education students", *Journal of Technology and Teacher Education*, Vol. 9 (3), pp. 321-247.
Askar, P. and Umay, A. (2001), " İlköğretim matematik öğretmenliği öğretmen adaylarının bilgisayarla ilgili öz-yeterlik algısı" ("Perceived computer literacy of the students in the elementary mathematics program"), *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, Vol.21, pp.1-8.
Bandura, A. (1986), *Social Foundations of Thought and Action: A Social Cognitive Theory,* , Prentice Hall, Englewood Cliffs, NJ.
Cassidy, S. and Eachus, P. (1998), " Developing the computer self-efficacy (CSE) scale; investigating the realtinship between CSE, gender and experience with computers", Computer Self-Efficacy Web Site, available at: www.chssc.salford.ac.uk/healthSci/selfeff/selfeff.htm (accessed May, 5, 2002)
Compeau, D.R. and Higgins, C.A. (1995), " Computer self-efficacy: development of a measure and initial test", *MIS Quarterly*, June, pp.189-211.
Delcourt,M. and Kinzie, M. (1993)," Computer technologies in teacher education: the measurement of attitudes and self-efficacy", *Journal of Research and Development in Education*, Vol.27,pp.31-7.
Karsten, R. and Roth, M.R. (1998), " The relationship of computer experience and computer self-efficacy to performance in introductory computer literacy courses", *Journal of Research on Technology Education*, Vol.31 No.1, pp. 14-24.
Kurbanoglu, S.S. (2003), " Self-efficacy: a concept closely linled to information literacy and lifelong learning", *Journal of Documentation*, Vol. 59 No.6, pp. 635-646.
Pajares, F. (1996) *Self-Efficacy Beliefs in Academic Settings, Review of Educational Research,* Winter 1996, Vol.66, No.4, pp. 543-578.
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 ComputingResearch*, Vol. 19 (4), pp367-381.
Usluel,Y. K. (2007).Can ICT usage make a difference on student teachers' information literacy self-efficacy, *Library and Information Science Research*, 29, 92-102.
Ramalingam, V., La Belle, D. and Wiedenbeck, S (2004) Self-effcacy and mental models in learning to program. ACM SIGCSE Bulletin, Volume 36- Issue 3, 171-175.
Zimmerman, B.J. (1995), "Self-efficacy and educational development", in Bandura, A. (Ed.), *Self-efficacy in Changing Societies*, Cambridge University Press, New York, NY, pp.202-231.