Informatics in Education, 2013, Vol. 12, No. 2, 207–219

Measuring Students' Acceptance and Confidence in Algorithms and Programming: The Impact of Engagement with CS on Greek Secondary Education

Spyros DOUKAKIS[1], Michail N. GIANNAKOS[2*], Christos KOILIAS[3], Panayiotis VLAMOS[4]

[1] The American College of Greece – Pierce
[2] Dept. of Computer and Information Science
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
[3] Dept. of Informatics, Technological Educational Institution (TEI) of Athens, Greece
[4] Dept. of Informatics, Ionian University, Corfu, Greece
e-mail: sdoukakis@acg.edu, michailg@idi.ntnu.no, ckoilias@teiath.gr, vlamos@ionio.gr

Received: March 2013

Abstract. This paper presents results of a questionnaire focused on investigating students' confidence and behavioral intention in the area of programming, particularly that of structures, problem solving, and programming commands (Conditional – Loop). Responses from 116 1st year students regarding informatics were used. The results indicate that the engagement with programming logic yields a positive impact on students' confidence and acceptance. In addition, all the measured factors are related relatively strongly. Our findings demonstrate that students' prior direction (at Lyceum) has a significant impact on their Confidence for using Programming Commands (CPC) and Confidence for using Data Structures (CDS); however, prior direction does not have any impact on learners Problem Solving Confidence (PSC) and Behavioral Intention (BI) for programming. In the conclusion, several issues regarding the courses of programming are discussed.

Keywords: adoption, algorithms, curriculum, data structures, Greece, programming, problem solving.

1. Introduction

For many years computing has been included in the curriculum as a distinct discipline in secondary education on a global scale. Computing focuses on how computers work (hardware) and how to program them (programming and software development), whereas ICT (Information and Communication Technology) is focused on how to use computers.

* This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007–2013) under grant agreement no 246016.

Specifically, programming with an emphasis on algorithms as a methodology of thought appears to be an important objective in students' education and aiming, among other things, in their preparation for tertiary education (Eliasson *et al.*, 2006; Wilson and Shrock, 2001).

Successful programming largely depends on its acceptance by potential university students or by high school students. Several models and theories have been used to address the issue of acceptance. The Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh *et al.*, 2003) is one of the most widely and successfully used. Other researchers have empirically explained (using UTAUT or its initial form of TAM) several issues regarding students' acceptance. In this light, we aim to measure empirically the acceptance of programming using the construct of Behavioral Intention (BI) adopted by UTAUT. We used four items to measure the BI construct, collected based on the literature review of empirical studies (Ngai *et al.*, 2007; Sánchez and Huero, 2010).

Additionally, the content of the curriculum largely defines the knowledge that students are expected to gain from a course in algorithms and programming. According to various curricula (Hubwieser *et al.*, 2011; Stephenson *et al.*, 2005; Tucker *et al.*, 2004), the purpose of a course in algorithms and programming is to foster analytical and synthetic thought in students, to develop their creativity and imagination in design, to foster their austerity and clarity of formulation, to develop their methodology skills, and finally to enable them to proceed to problem solving using programming knowledge (Vakali *et al.*, 1999).

In particular, it is essential for students, through a course in algorithms and programming, to be able to (Tucker *et al.*, 2004; Cassel *et al.*, 2008; Greek Pedagogical Institute, 1997):

A) Acquire knowledge regarding problem analysis.
B) Solve algorithmic problems using structures of sequence, conditional, loop.
C) Select data Structures when necessary.

The remainder of the paper is organized as follows: in the next section, we present the overview of the state of programming courses in Greece. In the third section, we give a short literature review of the framework of programming courses on other countries and the added value of these courses. The fourth section outlines the research hypotheses, followed by the applied research approach, as well as the datasets and respective measures. The sixth section presents the empirical findings derived, while the last section of the paper draws the conclusions of this research and discusses several implications and ideas for further research in the area.

2. Overview

In Greece, the teaching of Computing and ICT in secondary education is conducted by teachers holding at least an undergraduate degree in Computer Science, Computer Engineering, or Applied Informatics. Secondary Education in Greece is divided into two cycles: compulsory lower secondary and non-compulsory upper secondary education. Compulsory lower secondary education is provided in Gymnasium, while non-compulsory upper secondary education is provided in one of two types of schools: the General

Lyceum (GL) and Vocational Lyceum (EPAL). Parallel to these full-time day schools are evening secondary schools. The duration of studies in Gymnasium, both day and evening, is three (3) years. The duration of studies in a GL is three (3) years, unless the GL operates as an evening school, in which case duration is four (4) years. Computing and ICT courses are mandatory during Gymnasium years and aim to develop students' skills in the use of ICT (operating systems, word processing, spreadsheets, image processing etc). In the third year of Gymnasium, students acquire fundamental algorithmic and programming skills within a LOGO based environment. As part of a pilot programme, ICT courses have also been introduced in primary education for two teaching hours per week. In both day and evening GL, ICT has been taught as an elective or direction course since 1999. Finally, EPAL offers an Information Science sector with the following specialization: "System, Applications and Computer Networks' Support". Thus, besides mandatory education (primary and lower secondary), students in all the classes of Lyceum can select certain ICT from a wide range of various subjects. In the last two classes of General Lyceum, students select one of three directions, (technological, scientific, or theoretical). If students in the last grade select the technological direction, they attend a course in algorithms and programming, entitled "Application Development in a Programming Environment", for which they are assessed through national exams. Accordingly, in EPAL, if students select to specialize in ICT, they attend certain ICT courses. From these, they are assessed through national exams in the course of Structured Programming. The grade acquired in this examination is part of the consideration used in selecting students for admission in higher education programmes (Eyrydice, 2009).

Some of the goals of these two courses are the same. Both of them focus on the algorithmic approach and on the development of problem-solving skills in a programming environment; however, in the course of *Structured Programming*, students also deal with programming techniques and specific programming language (Pascal). Nevertheless, the overall aim of both courses is to develop analytical and synthetic thinking, acquire methodological skills, and be able to solve simple problems within a programming environment. Many basic algorithmic and programming concepts, such as conditions, expressions, and logical reasoning, are fundamentals of general knowledge and skills to be acquired in general education; most of these concepts are not included in other disciplines.

The curriculum states that this subject must be taught (at least partially) in a computer lab. The Greek Pedagogical Institute (Ministry of Education) has certified specific Educational Software to support the lab work, especially for the course of *Application Development in a Programming Environment*, while the course of *Structured Programming* features the environment of Pascal.

After the exams, on the basis of a) the total grades received, b) the courses examined, and c) their preferences, students enter tertiary education. Their admission is based upon certain rules, according to which their previous area of study is not the primary criterion for admission. Thus, it is possible for a student to be admitted in a Department of Informatics in tertiary education without having attended the course of either *Application Development in a Programming Environment* or *Structured Programming* accordingly.

The course of *Application Development in a Programming Environment* has been taught in secondary education since 1999, while the course of *Structured Programming* assessment at a national level has only recently started; however, no research has been conducted to draw conclusions contributing to the study of possible benefits from algorithmic courses, as well as the confidence of the students graduating from secondary education.

3. The International Context

Recent activity in countries such as USA, the United Kingdom, New Zealand, and South Korea display a growing awareness of the importance of rigorous computer science, specifically programming education in schools. Consequently, we have seen serious efforts to introduce or to improve programming in schools. Reports conducted by (1) the Association for Computing Machinery (ACM) and the Computer Science Teachers Association (Wilson *et al.*, 2010), and (2) ACM ITiCSE Working Group: Informatics in Secondary Education (Hubwieser *et al.*, 2011) revealed that CS courses face problems regarding their lack of exposure and motivators, which are quite essential. During the keynote presentation of the last International Conference on Informatics in Schools (ISSEP), the growing awareness of the importance of rigorous computer science and programming education became clear, as well as how this awareness can contribute to the successful, self-responsive, and self-deciding life in the modern world (Hubwieser, 2013).

Research in many countries, shows that students' prior engagement with programming and algorithmic courses results in better achievements in programming at tertiary education (Taylor and Mounfield, 1991, 1994; Hagan and Markham, 2000; Holden and Weeden, 2004). In the students' transition between secondary and tertiary education, it seems that students who possess experience in programming have significant advantages over the others (Clark and Boyle, 2005).

Cantwell-Wilson and Shrock, (2001) indicated that university students' "comfort level" in programming and computer science is highly related with their engagement with similar courses in their secondary education. In alignment with that, university students who have not participated in programming courses in secondary education perceive difficulty during the introductory courses of IT and CS department. Brown *et al.*, (1997) indicated that "students who find the course difficult are intimidated by seeing other students who have prior programming experience completing the assignments very quickly" (Brown *et al.*, 1997, p. 2).

On the same line, Dagiene and Jevsikova (2012) argue that the purpose of programming courses in secondary education is to motivate students to choose informatics in their further studies. As such, the highlight of Taylor and Mounfield's research (1989) regarding the precondition of programming course in order to help departments to maintain strong, healthy undergraduate programs is still timely. According to Ramalingam *et al.*, (2004) prior experience in programming contributes to increasing students' self-efficacy, in addition to affecting their perceptions for their capabilities.

Hence, the existence of the *Application Development in a Programming Environment* course in secondary education and the lack of relevant research motivate us to investigate the following research hypotheses regarding the impact of students' engagement with programming course on their secondary education.

4. Research Hypotheses

We carried out an experiential study aiming to explore issues regarding a) the benefits that students gain from relevant courses, b) their confidence after completing relevant courses, and c) the acceptance of programming.

Thus, we formulated the following hypotheses:

- *H1. Students' direction (technological or scientific) influences their behavioral intention (BI) to use programming.*
- *H2. Students' direction (technological or scientific) influences their level of problem solving Confidence (PSC).*
- *H3. Students' direction (technological or scientific) influences their confidence for using programming command (CPC).*
- *H4. Students' direction (technological or scientific) influences their confidence for using data structures (CDS).*

5. Research Approach

5.1. *Sampling*

Our research approach included a survey composed of questions on demographics and education information (gender, direction, kind of education, grades) of the sample and on the constructs of BI, PSC, CPC and CDS as defined in Table 2. Questionnaires were distributed in two different Departments of Informatics in Greece. The surveys were conducted during the first two weeks of October 2010, before the start of courses. The final sample of respondents consisted of 116 1st year students. From the total of respondents, 64 (55.2%) were males, and 19 (16.4%) were females. The vast majority, 94 (81.0%), of the respondents were educated in General Lyceum (GL), and 81 (69.8%) had followed the technological direction. In the Table 1 we present the detailed profiles of our sample.

Table 1
Students' demographic profile

Demographic Profile		No	%
Gender	Male	64	55.2%
	Female	19	16.4%
Kind of Lyceum	General Lyceum (GL)	94	81.0%
	General Evening Lyceum (GEL)	2	1.7%
	Vocational Lyceum (EPAL)	15	12.9%
	Vocational Evening Lyceum (EPAEL)	3	2.6%
Direction	Technological	81	69.8%
	Scientific	17	14.7%
	Theoretical	1	0.9%

* We have some missing fields in our data

5.2. *Measures*

A survey questionnaire was used in this study. It consisted of two sections; the first section required that participants provide their demographic and education information (gender, institution, direction, education), and the second section included items which measure the constructs of BI, CPS, CPC and CDS. Table 2 lists the questionnaire items adapted from previous studies to measure students' perceptions and confidence. For all variables, 5-point Likert scales were used to quantify the information.

6. Research Findings

Fornell and Larcker (1981) proposed three procedures to assess the convergent validity of any measure in a study: (1) composite reliability of each construct, (2) item reliability of the measure, and (3) the average variance extracted (AVE).

First, we carried out an analysis of composite reliability and dimensionality to check the validity of the scale used in the questionnaire. Regarding the reliability of the scales, Cronbach's (1951) α indicators' was applied and inter-item correlations statistics for the items of the variable. According to Fornell and Larcker (1981) and Hair, *et al.* (2006), Cronbach's α value greater than 0.7 indicates a high reliability. As we can see in Table 2, the result of the test revealed acceptable indices of internal consistency in the BI (0.914), PSC (0.785), CPC (0.947), and CDS (0.926) of the students.

At the next stage, we proceeded to evaluate the reliability of the measure. The reliability of an item was assessed by measuring its factor loading onto the underlying construct. All items exhibited factor loadings greater than 0.6, which is considered a good indicator of validity at the item level. The factor analysis identified four distinct factors; 1) Behavioral Intention (BI), 2) Problem Solving Confidence (PSC), 3) Confidence for using Programming Commands (Conditional – Loop) (CPC), and 4) Confidence for using Data Structures (CDS) (Table 2). Together, the four factors accounted for 72.504 percent of the total variance.

The third step for assessing the convergent validity is the average variance extracted (AVE); AVE measures the overall amount of variance that is attributed to the construct in relation to the amount of variance attributable to measurement error (Segars, 1997). Convergent validity is found to be adequate when the average variance extracted equals or exceeds 0.50 (Segars, 1997).

After the factor identification, we used the average mean value of the items of each factor in order to measured students' acceptance and confidence regarding the courses of programming. The levels of acceptance (BI=4.09/5) and confidence (PSC=3.79/5; CPC=3.97/5; CDS=3.49/5) are relatively high amongst surveyed students.

In the Figure 1 (Fig. 1), the recognized factors are presented.

Despite the fact that these factors arise from an orthogonal rotation and are separable in terms of item loadings, they are correlated (see Table 3). The Pearson correlations between the factors suggest that all the factors identified are related relatively strongly.

Table 2
Items and Summary of measurement scales

	Items	Mean	S.D.	Loadings	CR	AVE
Behavioral Intention	I plan to cope up with programming in the future. (BI1)	4.053	1.17	.843	.914	0.73
	I intent to continue programming in the future. (BI2)	4.129	1.17	.882		
	I intent to cope up with programming routinely. (BI3)	4.052	1.20	.866		
	I expect my use of programming to continue in the future. (BI4)	4.113	1.18	.830		
Conf	*I feel confident in…*					
Problem Solving Confidence	Fully understanding the problems presented to me (PSC1)	3.835	0.93	.701	.785	0.50
	Identifying the components of a problems (PSC2)	3.687	1.05	.769		
	Analyzing a problem to other simpler ones (PSC3)	3.948	0.98	.687		
	Identifying the data provided for handling a problem (PSC4)	3.974	0.93	.675		
	Posing a problem, formulating it accurately and completely (PSC5)	3.526	0.98	.683		
Confidence for Sequence – Conditional – Loop)	Formulating the forms of conditional statement if (CPC1)	4.115	1.23	.742	.947	0.62
	Discerning the differences of the forms of conditional statement if (CPC2)	4.158	1.17	.847		
	Selecting the best form of conditional statement depending on the problem (CPC3)	3.896	1.22	.784		
	Formulating the loop statement (CPC1)	4.122	1.17	.810		
	Selecting the best loop statement (CPC2)	3.783	1.21	.794		
	Using the appropriate loop statement (CPC3)	3.948	1.10	.791		
	Composing algorithms which use all three basic statements: sequence, conditional and loop (CPC4)	3.800	1.26	.715		
Confidence for Data Structures	Deciding whether it is necessary to use an array (CDS1)	3.478	1.32	.761	.926	0.64
	Selecting the formula of array (one-dimensional, two-dimensional, etc.) (CDS2)	3.644	1.42	.844		
	Entering, processing and printing the items of an array (CDS3)	3.687	1.38	.855		
	Doing general exercises and exercises of searching and sorting using the structure of the array (CDS4)	3.509	1.37	.793		
	Defining the structures of the stack and queue with the correspondent operations (CDS5)	3.113	1.39	.726		

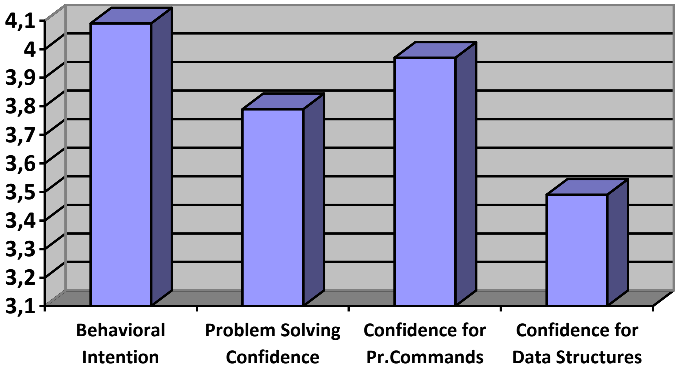

Fig. 1. Average mean value of each factor based on its items.

Table 3

Pearson correlations between Factors (n = 116)

	Behavioral Intention	Problem Solving Confidence	Confidence for using Programming Commands	Confidence for using Data Structures
Behavioral Intention	1			
Problem Solving Confidence	.247**	1		
Confidence for using Programming Commands	.455**	.447**	1	
Confidence for using Data Structures	.347**	.391**	.673**	1

**. Correlation is significant at the 0.01 level

To examine the hypotheses H1–H4, we used the t-test method, including the one dependent variable (direction) and four independent variables (BI, PSC, CPC, CDS). As we can see from the outcome data in Table 4, students' previous direction (in Lyceum) does not exhibit significant impact on BI, while PSC exhibits a highly significant impact on their CPC and CDS. This result provides support for hypotheses H3 and H4.

Table 4

Hypotheses testing using t-test method

	Mean (SD)		t	df	Sign.	Result
Direct.	Technological	Scientific				
BI	4.253 (0.95)	3.956 (1.07)	1.144	96	.255	i.d.
PSC	3.911 (0.67)	3.859 (0.68)	0.291	96	.772	i.d.
CPC	4.342 (0.71)	2.801 (1.24)	7.052	96	.000*	s.d.
CDS	3.919 (0.92)	2.459 (1.26)	5.552	96	.000*	s.d.

* p<0.001; i.d.=insignificant difference s.d.=significant difference

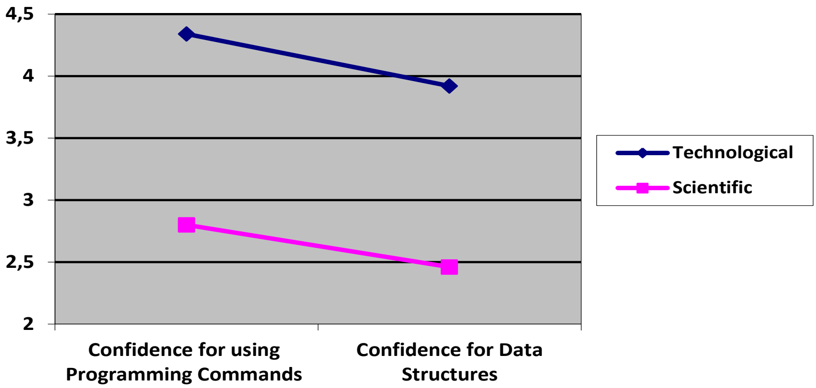

Fig. 2. The influence of direction in "Confidence for using Programming Commands" and "Confidence for Data Structures".

These results provide strong support for the notable influence of "Direction" in learners' confidence. The Figure 2 (Fig. 2) clearly exhibits the impact of direction on learners CPC and CDS.

7. Discussion

Computer science is one of the most evolving sciences, and its applications appear increasingly in everyday activities of people. In recent years, the acquisition of various stimuli and knowledge of the principles of computer science, as expressed in a variety of courses (Gal-Ezer and Harel, 1999; Tucker, 2010; Ragonis, 2009) has occurred among increasingly younger students. One reason is that recent years' students develop new representations different from those used in traditional classroom (Saeli *et al.*, 2010). Thus, the development of algorithmic thinking pursued in targeted courses in secondary education helps students to develop up to date skills and competences that may not be possible to achieve in other subjects (Soloway, 1993).

From the results it appears that secondary students who have obtained admission to a department of informatics of higher education and intend to deal with programming, state high confidence to solve problems, design algorithms, and use appropriate data structures when essential.

Moreover, students of tertiary education who have attended the course of algorithmic design as secondary students declare higher confidence than the students of tertiary education who have not enrolled in relevant courses. These findings match what Schulte and Magenheim (2005) found for students who previously attended computer literacy courses in secondary education; however, the different directions from which the students obtained admission to a department of informatics of higher education influences the teaching approach of the staff of the departments of informatics. As a result, it seems that the precondition of secondary education course on algorithmic and programming and the successful evaluation of this knowledge before entrance into CS and IT departments can highly improve the quality and the studies on these departments. This is in accordance with Taylor and Mounfield (1989), who have indicated that it might have positive benefits if the students obtained admission to a department of informatics with the condition that they have already acquired knowledge of design algorithms.

Another interesting finding is that the direction does not affect the confidence in problem solving (PSC) or Behavioral Intention (BI) in programming. Presumably, regardless of direction, most entrants declare that they want to study in a computer department of tertiary education because they wish to evolve in programming (Fogler and LeBlanc, 2007). Moreover, confidence in problem solving is developed by solving problems in various areas (mathematics, science, etc.), which is therefore not related to direction.

At the same time, it seems to be advantageous that the curriculum does not focus on a particular programming language during secondary education (Ginat, 2006). This highlights the fact that different departments of informatics have different curriculum, which possibly deepen to specific and different programming languages. The focus on algorithm development, without students' involvement with the techniques of programming languages, contribute to the development of critical thinking and problem solving skills, which has emerged globally as an important component of modern curricula.

8. Conclusions and Further Research

In this empirical study, we measured students' acceptance and confidence regarding the courses of programming. The levels of acceptance and confidence are relatively high amongst students. Results show the advantage gained by tertiary education students, whose curriculum includes programming courses, and who had attended courses of Algorithms and Programming as secondary education students. On the basis of the above, one could argue that attending relevant courses in secondary education is beneficial and contributes to achievement in higher level courses in tertiary education. Nevertheless, a) being with students without prior knowledge of algorithms and programming and b) the absence of a research that would provide information regarding the academic level of freshmen from the tertiary institution viewpoint prevent safe conclusions. This study yields new questions as to whether a) the relevant education at a secondary level, besides the confidence that it offers to students, also leads to better academic performance, b) it is preferable for students' training in algorithmic structures to be carried out in a common programming language rather than other ways of developing algorithms, and c) the knowledge that students acquired from secondary education is the kind of knowledge that will support their university education.

In Greek secondary education, there are several classes offering informatics and technology curricula. Concerning our future research program, we aim to conduct large scale pre – post studies in order to identify differences among students attending technology, informatics, and programming courses and those who do not. Our main target is to identify cognitive, performance and motivational differences affected by the involvement with informatics and programming courses.

References

Brown, J., Andreae, P., Biddle, R., Tempero, E. (1997). Women in introductory computer science: experience at Victoria University of Wellington. *SIGCSE Bulletin*, 29(1), 111–115.

Cassel, L., Clemens, A., Davis, G., Guzdial, M., McCauley, R., McGettrick, A., Sloan, B., Snyder, L., Tymann, P., Weide, B.W. (2008). *Computer Science Curriculum 2008: An Interim Revision of CS 2001: Report from the Interim Review Task Force*. ACM and IEEE Computer Society. `http://www.acm.org/education/curricula/ComputerScience2008.pdf`

Clark, M., Boyle, R. (2005). The transition from school to university: would prior study of computing help? In: Mittermeir, R., (Ed.), *Informatics in Secondary Schools: Evolution and Perspectives*, Berlin, Springer-Verlag, 37–45.

Cronbach, L.J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3), 297–334.

Dagiene, V., Jevsikova T. (2012). Reasoning on the content of informatics education for beginners. *Socialiniai Mokslai*, 4 (78), 84–90.

Eliasson, J., Westin, L.K., Nordstrom, M. (2006). Investigating students' confidence in programming and problem solving. *Frontiers in Education Conference*. 36th Annual Digital Object Identifier, 22–27.

Eyrydice (2009). *Organisation Of The Education System In Greece* 2008/09. `http://eacea.ec.europa.eu/education/eurydice`

Fogler, H.S, LeBlanc, E.S. (2007). *Strategies for Creative Problem Solving*. Prentice Hall.

Fornell, C., Larcker, D.F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research*, 48, 39–50.

Gal-Ezer, J., Harel, D. (1999). Curriculum and course syllabi for a high-school CS program. *Computer Science Education*, 9, 114–147.

Ginat, D. (2006). On novices' local views of algorithmic characteristics. In: Mittermeir, R.T. (Ed.), *ISSEP 2006*, LNCS, 4226, 127–137.

Greek Pedagogical Institute (1997). *Informatics Curriculum for Upper Secondary Education* (in Greek).

Hagan, D., Markham, S. (2000). Does it help to have some programming experience before beginning a computing degree program? In: *Proceedings of Integrating Technology into Computer Science Education Conference, ITiCSE-2000*. 25–28.

Hair, J.F., Jr., Black, W.C., Babin, B.J., Anderson, R.E., Tatham, R.L. (2006). *Multivariate Data Analysis*, 6th edn. Upper saddle River, NJ: Prentice-Hall International.

Hubwieser, P., (2013). The darmstadt model: a first step towards a research framework for computer science education in schools. In: Diethelm, I., Mittermeir, R.T. (Eds.), *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages. 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2013*. Oldenburg, Germany, 1–14.

Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M.N., Knobelsdorf, M., Magenheim, J., Mittermeir, R.T., Schubert, S.E. (2011). Computer science/informatics in secondary education. In: *Proc. of the 16th Annual Conference Reports on Innovation and Technology in Computer Science Education*. WG Reports, 19–38.

Ngai, E.W.T., Poon, J.K.L., Chan, Y.H.C. (2007). Empirical examination of the adoption of WebCT using TAM. *Computers and Education*, 48(2), 250–267.

Ragonis, N. (2009). Computing pre-university: secondary level computing curricula. In: Wah, B.W. (Ed.), *Wiley Encyclopedia of Computer Science and Engineering*, 5(1), 632–648.

Ramalingam, V., LaBelle, D., Wiedenbeck, S. (2004). Self-efficacy and mental models in learning to program. In: *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education*. ACM, NY, 171–175.

Saeli, M., Perrenet, J., Jochems, W., Zwaneveld, B. (2011). Teaching programming in secondary school: a pedagogical content knowledge perspective. *Informatics in Education*, 10(1), 73–88.

Sánchez, R.A., Huero A.D. (2010). Motivational factors that influence the acceptance of Moodle using TAM. *Computers in Human Behavior*, 26(6), 1632–1640.

Schulte, C., Magenheim, J. (2005). Novices' expectations and prior knowledge of software development: results of a study with high school students. In: *Proceedings of the first international workshop on Computing education research (ICER)*, 143–153.

Segars, A.H. (1997). Assessing the unidimensionality of measurement: a paradigm and illustration within the context of information systems research. *Omega International Journal of Management Science*, 25(1), 107–121.

Soloway, E. (1993). Should we teach students to program? *ACM Communications*, 36, 21–24.

Stephenson, C., Gal-Ezer, J., Haberman, B., Verno, A. (2005). The New Educational Imperative: Improving High School Computer Science Education. *Final Report of the CSTA Curriculum Improvement Task Force*.

Taylor, H., Mounfield, L. (1991). An analysis of success factors in college computer science: high school methodology is a key element. *Journal of Research on Computing in Education*, 24, 240–245.

Taylor, H., Mounfield, L. (1994). Exploration of the relationship between prior computing experience and gender on success in college computer science. *Journal of Educational Computing Research*, 11, 291–306.

Taylor, H., Mounfield, L. (1989). The effect of high school computer science, gender, and work on success in college computer science. *ACM SIGCSE Bulletin,* 21(1), 195–198.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., Verno, A. (2004). *A Model Curriculum for K-12 Computer Science*. The ACM K-12 Education Task Force.

Tucker, A. (2010). K-12 computer science: aspirations, realities and challenges. In: Hromkoviˇc, J., Královiˇc, R., Vahrenhold, J. (Eds.), *ISSEP 2010, LNCS, 5941*, 22–34.

Vakali, A., Giannopoulos, E., Ioannidis, N., Koilias, C., Malamas, K., Manolopoulos, Y., Politis, P. (1999). *Applications Development in a Programming Environment*. Pedagogical Institute, Ministry of Education, Athens, Greece, (in Greek).

Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D. (2003). User acceptance of information technology: toward a unified view. *MIS Quarterly*, 27(3), 425–478.

Wilson, B.C., Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. In: Russell, I. (Ed.), *The Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education*. In: *SIGCSE Bulletin inroads*, 33(1), 184–188.

Wilson, C., Sudol, L., Stephenson, C., Stehlik, M., (2010). *Running On Empty: The Failure To Teach K-12 Computer Science In The Digital Age*. Association for Computing Machinery. http://www.acm.org/runningonempty//fullreport.pdf

S. Doukakis received his PhD in Mathematics Education from the University of Aegean, Greece. He studied Mathematics and Informatics and holds three MSc degrees. He is a secondary teacher of Mathematics and Informatics at The American College of Greece-Pierce. His research interest included pupil learning with technology, student assessment and teacher's professional development.

M.N. Giannakos is an ERCIM/Marie Curie fellow in the Department of Computer and Information Science at Norwegian University of Science and Technology (NTNU) and a Visiting Richard T. Cheng Fellow in the Center for Real-time Computing, Virginia, USA. He earned a Ph.D. in Educational Technology from Ionian University, Corfu, Greece. Giannakos is the main author of more than thirty five peer-reviewed research articles in several international journals and conferences in the area of Educational Technology and Computer Science Education. He has worked at several research projects funded by diverse sources like EC, Microsoft Research and NSF.

Ch. Koilias is a Professor in Technological Educational Institution (TEI) of Athens, Department of Informatics since 1985. He studied Mathematics and received his MSc at the University Pierre et Marie Curie (PARIS VI) and his PhD at the University of Aegean in Information Systems. His main research interests are data structures and quantitative statistical analysis. He has written more than 40 books in information technology. He is an active member of Greek Computer Society.

P. Vlamos is an Associate Professor in the Department of Informatics at the Ionian University, Corfu, Greece. He received his Diploma in Mathematics from the University of Athens (1991) and his Ph.D. degree in Mathematical Modeling from the National Technical University of Athens, Greece, (1997). His research interests mainly concern: Informatics in Education, Mathematical Software towards Education and Mathematical Modeling and Simulation.

Studentų palankumo algoritmavimo ir programavimo mokymui rodiklių matavimas: informatikos mokymo Graikijos vidurinėje mokykloje įtaka

Spyros DOUKAKIS, Michail N. GIANNAKOS, Christos KOILIAS,
Panayiotis VLAMOS

Straipsnyje pristatomi apklausos, skirtos ištirti studentų pasitikėjimą ir elgesį mokantis pagrindinių programavimo komandų ir struktūrų rezultatai. Buvo apklausta 116 pirmųjų studijų metų studentų. Rezultatai rodo, kad mokinių supažindinimas su programavimo logika teigiamai veikia studentų pasitikėjimą ir palankumą šiam kursui. Visi matuoti faktoriai yra stipriai susiję. Gauti rezultatai rodo, kad ankstesnė studento mokymosi kryptis vidurinėje mokykloje turi reikšmingą įtaką jo pasitikėjimui naudoti programavimo komandas ir duomenų struktūras. Vis dėlto ankstesnis mokymasis neturi jokios įtakos studentų pasitikėjimui sprendžiant uždavinius ir elgesiui. Apibendrinant tyrimo rezultatus aptariama keletas programavimo kursų problemų.