Assessment of Self-regulated Attitudes and Behaviors of Introductory Programming Students

Ana Paula Ambrosio
Computer Science Institute
Federal University of Goias
Goiania, GO, Brazil

Scheila Martins
Centre for Informatics and Systems
University of Coimbra and Univ. da Região de Joinville
Coimbra, Portugal and Joinville, Brazil

Leandro Almeida, Amanda Franco
Institute of Education
University of Minho
Braga, Portugal

Fouad Georges
Technology Area
University Salgado de Oliveira
Goiania, GO, Brazil

***Abstract* - Great emphasis has been given to the cognitive and methodological aspects of learning and teaching programming. Nevertheless, research shows a significant correlation between student self-regulation and academic achievement in higher education. The Attitudes and Study Behaviors Inventory (IACHE) is a multidimensional inventory that integrates in its design behavioral, affective-motivational and cognitive aspects of learning. This paper presents the results obtained from the application of the IACHE inventory to 190 introductory programming students from Brazil and Portugal. Data suggests that Brazilian and Portuguese students have very similar attitudes and behaviors regarding learning strategy, self-efficacy perception and study activities' organization, which seems to suggest that the same type of students choose undergraduate computing programs in both countries.**

Keywords – Self-regulation; Behavior; Motivation; Cognition; Achievement; Programming, IACHE.

I. Introduction

Teaching basic programming in higher education is one of the greatest challenges of computing. Many students entering computing degree programs have their first contact with programming in their first semester, and are faced with the need to develop a logical/algorithmic reasoning, to learn about a new technology and acquire knowledge about a new language's syntax and semantics, in addition to dealing with their transition from high school to university. Given this scenario, it is not a complete surprise that these courses have been globally responsible for large failure and dropout rates in undergraduate computing programs [1].

Although great emphasis has been given to the cognitive and methodological aspects of learning and teaching programming, research shows a significant correlation between students' self-regulation and academic achievement in higher education. According to Zimmerman [2], self-regulation implies personal knowledge and motivation to make deliberate use of skills to act. In school, self-regulated learning involves the regulation of three main aspects of academic learning: *behavior*, which involves the active control of the many resources that students have available to them, such as their time and study environment; *motivation*, responsible for controlling and changing motivational beliefs such as self-efficacy and goal orientation, so that students can adapt personal resources to their school settings' demands; and *cognition*, which involves the control of several cognitive strategies implied in the learning process, such as the use of deep processing strategies [3]. In the following, we will focus on how each one of these regulatory dimensions of academic learning is related to students' academic achievement.

(i) *Behavior*. Starting in elementary school and throughout schooling, students face several tasks, contexts and learning methodologies that help them develop their study behavior, helping them find a learning approach that best fits their personality, beliefs and values. They set goals to be achieved, select resources, and implement strategies to tackle specific learning tasks [2]. In this context, study skills and academic learning approaches include, for example, time management, use of information resources, taking notes in class, communicating with teachers, or even, preparing for and taking examinations. In their meta-analysis, Hattie, Biggs and Purdie [4] conclude that if we aim to improve academic achievement and increase academic success in higher education, institutions must foster interventions directed at students' learning strategies.

(ii) *Motivation*. Understanding motivation to learn requires a deep analysis of the socio-cognitive components associated to personality (identification with the institution, chosen program, vocational project and career, accessibility to teaching, analysis of school success) and the quality of relationships in life environments (social identification and learning approaches) [5]. The motivational dimension has a great impact on the individual's cognitive development and is a determinant factor

978-1-4673-1352-0/12/$31.00 ©2012 IEEE

for success in the learning process [6]. It is motivation that fosters in the student the disposition to want to progress and reach the goals that were set, maintaining an adequate level of volition to overcome the demands that are being dealt with [2].

(iii) *Cognition*. The interaction between motivation and cognition positively impacts students' academic performance [7]. Furthermore, students who have a more adequate approach to learning and who are more motivated, elaborate more positive cognitions regarding themselves in the long term, increasing their chances of academic success [2]. In other words, such students build self-efficacy beliefs, which are influenced by their motivation and affect their judgment about personal ability to perform a task in a specific domain. In learning situations, self-efficacy influences the use of efficient cognitive strategies while solving problems, the amount of effort expended, the type of coping strategies adopted, the level of persistence in the face of failure, and the performance outcomes [8].

Based on these research findings, the "Attitudes and Study Behaviors Inventory" – *IACHE* (in Portuguese, *Inventário de Atitudes e Comportamentos Habituais de Estudo*) [9; 10], a self-report questionnaire, was developed by a Portuguese research group to assess behavioral, motivational and cognitive factors related to students' study habits and learning approaches in higher education.

Considering the difficulties, failure and dropout rates experienced by introductory programming students, and having in mind the impact of self-regulated attitudes and behaviors in academic achievement, we have used IACHE to assess Brazilian and Portuguese university students. Despite the clear differences between both countries, the programming learning-teaching scenario has some aspects in common, namely high failure and dropout rates, hence the interest in analyzing what students from Brazil and Portugal share in common and what sets them apart, in order to identify possible clues to successfully intervene on these issues.

II. Brazil and Portugal, Two Realities

Content delivered in introductory programming courses is very similar in Brazil and Portugal, despite the realities inside the classroom in both countries being very distinct. In the last few years, the Federal University of Goiás, in Brazil, and the University of Coimbra, in Portugal, changed the way introductory programming courses had been historically taught in their institutions. The reasons that have led to such changes varied, as well as the new chosen approaches, as presented in the following.

A. Federal University of Goiás (Brazil)

In 2008, Problem-Based Learning (PBL) with tablet PC support was introduced in the Computer Programming 1 course of the Bachelor in Computer Science's program of the Computer Science Institute of the Federal University of Goiás (UFG), in Brazil, to minimize the recurring problems of students' low achievement, high dropout rates and lack of motivation [11].

The forty students that enter the program are divided in two groups of twenty, so that each student can have his/her own tablet PC. The laboratory used to teach the course is composed of four trapezoid-shaped tables with up to six students each. Each student is provided with a tablet PC with the necessary environments installed. Besides the professors, one for each class, there is a teaching assistant for each class, usually an undergraduate student, to help during class and in specified hours after class, with a weekly workload of 12 hours.

In the classroom, the PBL method [12] is used to introduce the concepts in the course syllabus as a series of open-ended problems, using a method adapted from Nuutila, Torma and Malmi [13], according to which groups of four to five students work collaboratively to reach a solution to the proposed problems. The description of a problem is given to each group; students are given approximately 40 minutes to discuss among themselves possible ways to solve the problem, relating it with the knowledge they already possess and identifying topics for which they need more information or with which they are not yet familiar. After this "brainstorming stage", students filter their ideas and the group identifies learning objectives that represent knowledge they must have to solve the problem and that must be researched/studied. Outside the classroom, students work individually each of the learning objectives, and are not allowed to divide their task. Having obtained the necessary knowledge to solve the problem, students gather again to propose a joint solution, obtained from suggestions and individual solutions from each one of the members of the group. The resulting algorithm is then implemented. This process can take a week or more, depending on the complexity of the problem. Eventually the teacher can give a lecture addressing issues that were misunderstood by the students or to complement the learning objectives proposed by the groups.

The course is divided in two parts: the first using only the SICAS environment [14] for flowchart diagramming, and the second using the DevC++ environment. The SICAS environment allows students to draw flowcharts that are then automatically translated to C or Java. The system offers a drag and drop environment with constraints that guide the students in the algorithmic definition of the problem, contributing to the learning process. All basic programming concepts are discussed in the first phase, and again in the second phase. The first part extends for about a month and a half, while the second part lasts 3 ½ months. In each stage, several distinct problems are proposed. Examples of these problems include defining a calculator or implementing games such as Battleship and Pac Man.

This course is offered in the first semester of the program, with a total of 64 hours in the classroom (32 encounters of 2 hours, twice a week). Attendance is mandatory. Students that are absent in more than 25% of the classes fail the course. To be approved, the students must achieve a minimum of 5 points, in a 0-10 scale; these points are distributed by the activities of the course, and the teacher has autonomy to define the distribution. In the last few years, the distribution that is being used is: 20% for individual participation in class and exercises, 20% for group solution of problems, 30% for the mid-term exam, and 30% for the end-term exam.

B. University of Coimbra (Portugal)

In Europe, one of the most important recent events in the educational sphere has been the evolution and adaptation of universities to the Bologna Process, which led to the creation of a European Higher Education Area (EHEA) with unified strategies and development goals [15]. Within this new reality, the University of Coimbra (UC) has adapted its programs to conform to Bologna Declaration since 2005 [16]. In some cases, this means classes with a greater number of students and the standardization of some courses so they can be attended by students from several programs, which accounts for an expressive heterogeneity of student profiles in some of them. This is the case for the introductory programming disciplines. Students from the Bachelor's Degree in "Engineering and Information Technology" – LEI, in the Faculty of Science and Technology of the University of Coimbra (FCTUC), attend classes where the number of enrolled students is usually very high: between 200 to 300 candidates, who come from several programs such as "Industrial Engineering and Management" and "Communication and Multimedia".

In the new curriculum, introductory programming was divided in two courses: "Introduction to Programming and Problem-Solving" and "Principles of Procedural Programming" – PPP. The first was proposed as a transition programming course designed to present the first programming concepts to the students in a problem solving context using Phyton. This course if followed by "Principles of Procedural Programming" – PPP, the second programming course of the LEI program, aiming to support the basic problem solving knowledge acquired in the first semester which is taught using ANSI-C. There is no pre-requisite, which explains the high number of students enrolled in these courses.

Although the PPP course accepts students from several programs, the whole course was designed to fit the LEI students' profile. The program includes basic programming knowledge for understanding the ANSI-C programming language (1999 standard), memory management, pointers and algorithms for fundamental data structures. The course is based on content presentation lectures and more applied classes, where students can practice what they have learnt from lectures, with evaluation points done using small practical programming challenges done in pairs. Classes in the UC are usually organized into:

(i) Lectures (2 hours). All students enrolled in PPP have class with a coordinating professor responsible for the subject;

(ii) Practical sessions (2 hours). The class is divided in smaller groups in order to do hands-on lab exercises under the supervision of a teacher, who may or may not be the lecturer, depending on the total number of enrolled students and the capacity of the laboratories. These groups have usually 20 to 30 students, with a total of 10 to 14 groups for each course, and involve 3 to 4 teachers, in addition to the coordinating professor;

(iii) Practice-Lab Sessions (2 hours). These are not classes *per se*, but support sessions in which the students have study guides, reinforcements and clarification of doubts, with the aid of a tutor, usually a graduate student.

With the exception of practice-lab sessions, which are not mandatory, all classes require advance registration and attendance, except from working students. Besides classes, the activities developed during the course included: 10 exercise lists to be implemented in the practical sessions (with the designation of some exercises to be presented orally), a theoretical evaluation, a mini-project and a final exam taken at the end of the course. All activities are scored, and students must achieve a minimum of 10 points in a 0-20 scale, in order to be approved in that subject. To qualify for the final exam, each student must obtain a minimum frequency and grade that is equivalent to 35% of the points attributed to the activities. For those that do not succeed in the final exam, there is a second chance with a "special season exam". However, even amongst those who have the right to take the final exam, there are approximately 40% that choose not to take the exam because almost all exams occur in the same period and the students choose to study for those that they believe they have a better chance of success in. Furthermore, many students who failed "Introduction to Programming and Problem-Solving" enroll for PPP, as there is no pre-requisite and enrolment is done once a year. However, most of these students dropout in the first month, or right after the first exam is taken.

III. Goals of our Study

With our study, we aimed to use IACHE to assess Brazilian and Portuguese university students. We believe that the analysis of data obtained from IACHE can provide some important information about the attitudes and behavior of students, individually and in groups, and shed some light on cognitive and motivational aspects of programming students' profile.

Our main research goals were to:

(i) Apply IACHE to introductory programming students in Brazil and Portugal and estimate the psychometric proprieties of this instrument;

(ii) Identify possible differences between the profile of introductory programming students in Brazil and Portugal;

IV. Method

A. Participants

A total of 190 students – 72 from UFG (in Brazil) and 118 from UC (in Portugal) – participated in our study. To have a more homogeneous profile of our sample, we chose only UC students enrolled in PPP that come from the LEI program, as this program has a similar objective and structure as the Computer Science program of UFG. Furthermore, we chose to evaluate students from the PPP course that occurs in the second semester, as the course uses an imperative language, similar to the one used by the UFG students.

In UFG, testing was done in 2009-2, 2010-1 and 2010-2, always at the end of the course. In the Brazilian sample, 23 students were assessed in 2009-2, 31 in 2010-1 and 18 in 2010-2. On the whole, 63 were male and 3 were female (6 did not identify their sex). The sample ranged in age from 16 to 26 years (M = 18.5; SD = 1.68). IACHE was administered to students who had just completed or were completing the first semester of introductory programming.

In UC, testing was done in the beginning of the second semester of the 2008/2009 school year. From a total of 320 enrolled students in the PPP course, 244 were from the LEI program of which 118 answered the inventory. Of these, 109 were male and 9 were female. The sample ranged in age from 18 to 31 years (M = 20.0; SD = 2.73).

B. Procedures

IACHE was applied collectively, in a classroom environment. Students were informed about the study purposes and assured about the confidentiality of their results. Participation was voluntary and there was no financial compensation.

C. Instrument

The instrument used in our study, IACHE – "Attitudes and Study Behaviors Inventory" –, was administered to assess behavioral, motivational and cognitive aspects of students' study habits and learning approaches. This multidimensional inventory integrates in its design three mains factors: *behavior* (concerning actions, daily routines, time management and study materials), *motivation* (concerning commitment, interest, involvement and progress in the study) and *cognition* (concerning personal perceptions, strategies or approaches to learning). Regarding the learning approach, IACHE allows to contrast between a superficial and a comprehensive approach to learning: the first focuses on the memorization of information; the latter emphasizes a more significant learning and understanding of learned contents. According to these factors, IACHE is divided in five sub-scales:

- **Comprehensive Learning** - using reflection and analysis in depth of content, which means more effort and time spent by the student in learning, who is focused on understanding (10 items, $\alpha = .86$);
- **Surface Learning** - tendency to spend a minimal effort to learn. Learning is superficial, based on memorization and reproduction of contents (8 items, $\alpha = .81$);
- **Intrinsic Motivation/Involvement** - the availability for study activities, primarily related to requirements of intrinsic motivation (8 items, $\alpha = .83$);
- **Study Activities Organization** - examines the evidence of structured activities and study. Focus on how students organize and manage their study (time, materials, etc.) (10 items, $\alpha = .83$);
- **Self-efficacy Perception** - personal perceptions about capacity to succeed in the academic tasks, self-concept, expectations, etc. (8 items, $\alpha = .80$).

IACHE is composed of 44 items, distributed in the five dimensions previously described. Items are presented in a six-point Likert scale, according to the degree of agreement (1 = "never" and 6 = "always"). In the parenthesis associated to each dimension description above, the distribution of these items can be found, along with the value of Cronbach's alpha, used as a measure of the internal consistency or reliability of a psychometric test score for a sample of examinees, associated to each dimension. Values above .80 indicate good reliability. According to preliminary studies (see [10]), the factorial analysis and internal consistency of the items of IACHE have shown satisfactory results (Cronbach's alpha coefficient ranging between .80 and .86).

V. RESULTS

Before comparing the results obtained in the sample from Brazil and Portugal using IACHE, it was necessary to test the validity of the scale. This was undertaken through a factor analysis of the internal structure, in order to see if the items grouped alike in the two countries. Even though theoretical background, as well as preliminary studies of IACHE, assumed the existence of five dimensions, the factor analysis of our data captured three dimensions that can be interpreted with a minimum of coherence. Each one of the factors attained the 5% minimum variance criteria (Kaizer criteria), explaining together a total of 45.7% of the variance. Thus, there were identified three dimensions explaining what is common to both countries, which was used to test the hypothesis of the existence of differences between Brazilian and Portuguese students.

The items that did not meet a minimum of 0.5 factor loading criteria were eliminated. A closer analysis of the items associated to the three identified factors revealed that they correspond to the items originally associated to the *Study Activities Organization*, *Comprehensive Learning* and *Self-efficacy Perception* sub-scales proposed by the theory underlying IACHE. Hence, although the original factor analysis did not replicate the theory, some level of validity may be given to these three factors.

Following, a score per subject was calculated for each one of the three dimensions. This was done using a regression factor analysis. An original Z score was obtained and transformed to a T score, to avoid working with negative values. No correlation between the three dimensions was found according to the Varimax rotation factor that was implemented. A UNIANOVA was calculated for each dimension, separating the groups (Brazil and Portugal), with sex and age as covariates. Significant variance was found for the *Study Activities Organization* dimension ($F(1,188) = 4.64$, $p < .05$).

To better understand the variance encountered in *Study Activities Organization*, an analysis of the items associated to this dimension was undertaken using an Independent t-Test. Significant variance was found in item 17 ($t(93.35) = 5.49$, $p<.001$). This item asked if the student went regularly to the library to read or browse for books and documents. Brazilian students (M = 3.0; SD = 1.66) presented a higher mean than Portuguese students (M = 1.9; SD = 0.84), suggesting that Brazilian students go more often to the library to read or browse for books and documents, even though their mean was positioned in an intermediate level of the scale ranging from 1 to 6, corresponding 6 to maximum agreement. For the remaining items, no significant variance was found, with means ranging from 2.6 to 3.7 for both Brazilian and Portuguese samples, indicating that students rate themselves in an intermediate level of the scale.

VI. Other Considerations

In addition to the analysis of the IACHE results, we have undertaken an analysis of the academic grades obtained by both samples. Table I compares Brazilian and Portuguese realities, presenting the total number of enrolled students in the cohorts where IACHE was administered, the number of dropouts, as well as the number of students who failed even though attending classes (were not dropouts), and finally, those who succeeded. A mean grade was calculated for those who succeeded. A general mean grade was not available because in UC failure grades are not recorded. As the grade range in UC varies from 0 to 20, we have divided the mean grade of approved students (M = 14.12) by two, so that it can be compared to the Brazilian sample. Furthermore, we calculated the total percentage of approved and retained students considering only those who did not abandon the course.

TABLE I
Characterization of Programming Course Students in the Brazilian and Portuguese Samples

	Course-Program-University	
	PC1-INF-UFG	PPP-LEI-UC
Enrolled students	n = 110	n = 244
Dropout	n = 15 (14%)	n = 126 (52%)
Failed	n = 20 (18%)	n = 37 (15%)
Approved	n = 75 (68%)	n = 81 (33%)
Mean Grade of approved students	6.54	7.06
Not considering dropout students:		
Percentage of approved	79%	69%
Percentage of retained	21%	31%

According to the information presented in Table I, the teaching methodology appears to have a relevant impact on the dropout rate. However, it does not seem to have the same impact on the students' grade nor in the percentage of retained students (only 10 % difference). This may suggest that the methodology, by itself, is not decisive to reduce the percentage of failure. This may lead us to question the validity of the investment made by INF-UFG, where the number of students in each cohort was drastically reduced, given the expectation that a more personalized methodology would foster higher success rates.

This information, along with the results of our study, lead us to wonder if the high failure rates found in Introductory Programming courses may be mainly due to internal factors associated to the cognitive and motivational processes of the students. In fact, authors have suggested that the difficulties experienced by novice students who are learning to program are related to the lack of fitter mental models [17, 18, 19, 20, 21, 22]. For several reasons, including the difficulty of first year students, several researchers propose the development of Computational Thinking skills, i.e., thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent [23], as early as elementary school [24, 25]. However, these are hypothesis that would need to be tested in future studies.

VII. Conclusions

In the traditional method of teaching programming, students hardly feel excited because they have to concentrate on coding and compiling problems generated by the rigidity of professional programming languages, in addition to solving the algorithmic problem. In the aim of tackling such challenges, different approaches have been defined, mostly through the use of new classroom methodologies and the use of different programming languages and tools.

At UFG (in Brazil), the Computer Science program adopted, in 2008, the PBL method for teaching their CS1 course, along with the use of tablets PC and flowcharts. The results obtained were quite satisfactory. Although there had been no significant increase in average scores, compared to classes in previous years, there was a significant decrease in the number of failures and dropouts rates. However, there is still a 25% failure rate in this course.

The adaptation to the Bologna process undertaken in UC (in Portugal) aimed to make students more independent and proactive in building their own knowledge, returning teachers to their role of guides of the students' learning journey. Even though the idea is interesting, its implementation still has some issues to overcome, given the large classes and the heterogeneity of students' profile, and also the expressive dropout and failure rates.

To improve the CS1 outcomes, new solutions must be thought of, taking into account the students' profile, their attitudes towards learning and their method and strategies to acquire knowledge. In this sense, a first step is to better understand the students and their profile. To do so, we have applied the IACHE questionnaire to Brazilian and Portuguese university students, aiming to identify their attitudes and behavior towards study and academic learning. Data obtained from our application allowed us to observe that there are no major differences between Brazilian and Portuguese students concerning their attitudes and behaviors, despite the fact that they belong to two very idiosyncratic realities, where two very distinctive teaching approaches are being implemented.

An analysis of the academic outcomes of Brazilian and Portuguese students seems to indicate that the teaching methodology has an impact on the dropout rate, but not so much in the grades or in the number of retained students. This raises the hypothesis that the problems faced by students may be related to internal factors associated to their cognitive processes. Further investigation is needed to test such possibility, as well as to better comprehend and intervene on difficulties experienced in introductory programming courses, as well as to verify if a specific student profile can be expected when we consider students who chose Computer Science programs.

References

[1] P. J. Denning, "The field of programmers myth". Communications of the ACM, 47 (7), pp. 15-20, 2004.

[2] B. J. Zimmerman, "Becoming a self-regulated learner: An overview", Theory into Practice, 41, pp. 64-70, June 2002.

[3] B. J. Zimmerman, "A social cognitive view of self-regulated academic learning". Journal of Educational Psychology, 81, pp. 329-339, 1989.

[4] J. Hattie, J. Biggs, and N. Purdie, "Effects of learning skills interventions on student learning: A meta-analysis". Review of Educational Research, 66, pp. 99–136, 1996.

[5] M. V. Abreu, Cinco ensaios sobre a motivação humana (2nd ed.). Coimbra, Portugal: Livraria Almedina, 2002.

[6] A. H. Maslow, Motivation and personality (1st ed). NY: Harper, 1954.

[7] D. H. Schunk and B. J. Zimmerman, Self-regulation of learning and performance: Issues and educational applications. Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.

[8] A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change". Psychological Review, 84, pp. 191-215, 1977.

[9] S. Monteiro, R. Vasconcelos, and L. Almeida, "Rendimento acadêmico: Influência dos métodos de estudo". In *VIII Actas do Congresso Galaico-Português de Psicopedagogia*, pp. 3505–3516. Centro de Investigaçao em Educaçao do Instituto de Educaçao e Psicologia da Universidade do Minho, 2005.

[10] J. Tavares, L. Almeida, R. Vasconcelos, and J. Bessa, Inventário de Atitudes e Comportamentos Habituais de Estudo. Braga, Portugal: Universidade do Minho, 2003.

[11] A.P. Ambrósio and F. M. Costa, "Evaluating the impact of PBL and tablet PCs in an algorithms and computer programming course". In ACM Technical Symposium on Computer Science Education, 2010, Milwaukee-WI, 1, pp. 495-499. Proceedings of The 41st ACM Technical Symposium on Computer Science Education. New York: ACM Press, 2010.

[12] H. G. Schmidt. "Problem-based learning: Rational and description". Medical Education, 17, pp. 11–16, 1983.

[13] E. Nuutila, S. Torma, and L. Malmi, "PBL and Computer Programming – The Seven Steps Method with Adaptations." Computer Science Education, 15 (2), pp.123–142, 2005.

[14] C. Areias & A. Mendes, "A tool to help students to develop programming skills". In *Proceedings of the 2007 international conference on Computer systems and technologies*, CompSysTech '07, pages 89:1–89:7, New York,NY, USA. ACM. 2007.

[15] AREA, T. E. H. E. (1999). The Bologna Declaration of 19 june 1999.

[16] B. W¨achter, "The Bologna process: Developments and prospects". European Journal of Education, 39 (3), pp. 265–273, 2004.

[17] D. Ackermann and J. Stelovsky, "The role of mental models in programming: From experiments to requirements for an interactive system". Lecture Notes in Computer Science, 282, pp. 53-69, 1987.

[18] J. J. Cañas, M. T. Bajo, and P. Gonzalvo, "Mental models and computer programming". International Journal of Human-Computer Studies, 40 (5), pp. 795-811, 1994.

[19] S. Dehnadi, "Testing programming aptitude". In P. Romero, J. Good, E. A. Chaparro, & S. Bryant (Eds.), *Proceedings of the 18th Workshop of the Psychology of Programming Interest Group*. University of Sussex, pp. 22-37, 2006.

[20] R. E. Mayer, "The psychology of how novices learn computer programming". In E. Soloway, & J. C. Spohrer (Eds.), *Studying the novice programmer*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1989.

[21] D. N. Perkins, S. Schwartz, and R. Simmons, "Instructional strategies for the problems of novice programmers". In R. E. Mayer (Ed.), *Teaching and learning computer programming.* Hillsdale, NJ: Lawrence Erlbaum Associates, pp. 153-178, 1988.

[22] S. Wiedenbeck, D. LaBelle, and V. Kain, "Factors affecting course outcomes in introductory programming". In E. Dunican, & T. R. Green (Eds.), *Proceedings of the 16th Workshop of the Psychology of Programming Interest Group*. Carlow, Ireland: Institute of Technology Carlow, pp. 97-110, 2004.

[23] J. Cuny, L. Snyder, and J. Wing, "Demystifying computational thinking for non-computer scientists", work in progress, Retrieved April 2nd, 2012, from *www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf*.

[24] J. J. Lu and G. H. L. Fletcher, "Thinking about computational thinking", SIGCSE'09, Chattanooga, Tennessee, USA, March 3–7, 2009.

[25] C. L. L. Prottsman, "Computational thinking and women in computer science", MSc in the Department of Computer and Information Science University of Oregon Graduate School, 2011.