Work in Progress: Using Second Language Acquisition Techniques to Teach Programming - Results from a Two-Year Project

Dr. Christina Frederick, Embry-Riddle Aeronautical University, Daytona Beach

Dr. Frederick is currently a Professor and Graduate Program Coordinator in the Human Factors and Systems Department at Embry-Riddle Aeronautical University in Daytona Beach, Florida. Dr. Frederick received her Ph.D. in 1991 from the University of Rochester with a major in Psychological Development. She previously taught at the University of Rochester, Southern Utah University and the University of Central Florida. In 2000, Dr. Frederick joined the Human Factors and Systems Department at Embry-Riddle, where her work focused on applied motivation and human factors issues in aviation/aerospace. Dr. Frederick also served in various roles in University administration between 2004-2012, including Vice President for Academics and Research. Dr. Frederick's current research interests examine how individual differences interact with technology to enhance educational engagement and performance. Dr. Frederick is the author of more than 50 research publications, 4 book chapters and over 60 regional, national and international conference presentations on a wide range of topics in human factors and psychology. She is active in a number of professional associations, and is a Consultant for Psi Chi, the National Honor Society in Psychology.

Dr. Lulu Sun Ph.D., Embry-Riddle Aeronautical University, Daytona Beach

Lulu Sun is an associate professor in the Engineering Fundamentals Department at Embry-Riddle Aeronautical University, where she has taught since 2006. She received her Ph.D. degree in Mechanical Engineering from University of California, Riverside, in 2006. Before joining Embry-riddle, she worked in the consulting firm of Arup at Los Angeles office as a fire engineer. Her research interests include second language acquisition in programming languages, flipped classrooms, and distance learning. She is a member of the American Society for Engineering Education.

Work in Progress: Using Second Language Acquisition Techniques to Teach Programming. Results from a Two-Year Project

Christina M. Frederick and Lulu Sun

Embry-Riddle Aeronautical University

Abstract

This WIP paper presents two years of findings for an NSF funded project under the Research Initiation Grant in Engineering Education (RIGEE) program. The project (SLA-aBLe) is currently in the second year of implementation and assessment. Final results from the project will be presented and discussed at the annual ASEE conference. The project used second language acquisition (SLA) theory and techniques to facilitate learning in an introductory programming language class. The project was developed by a multi-disciplinary team and involved multiple instructors and sections of an introductory programming language class using MATLAB. Each semester, instructors trained in SLA techniques taught sections of both the SLA-aBLe and non-SLA-aBLe programming language course, and the performance of students in the different type of course sections was compared. Assessment of effectiveness was conducted in a scientifically rigorous and extensive manner, using multiple surveys, student grades and instructor assessment. Results from the first year of implementation indicated that students in the SLA-aBLe sections of the programming class exhibited higher end of course lab scores, exam scores and grades than students in non-SLA-aBLe sections of the same course. In addition, students in the SLA-aBLe sections reported higher levels of motivation and less frustration than students in the non-SLA-aBLe sections of the class. Perceptions of faculty competence did not differ by type of course section or across faculty teaching the class. This project is continuing into its final year of implementation during the 2016-2017 academic year. Researchers will continue to assess the course using student perceptions, and class outcomes to determine effectiveness of the program. The proposed paper will focus on presenting two years of data from the project, including discussion of the overall success of using SLA techniques in engineering education.

Introduction

The requirement for engineering and computer science students to learn a programming language is standard practice, however teaching undergraduates a programming language can be challenging[1-5]. One reason for this is that undergraduate students find learning a programming language to be difficult, especially if they were not exposed to programming languages previously[2, 5]. A programming course is typically required in the first year of study and students learning a programming language for the first time experience it as a complex task, in which they have to use logical reasoning and problem solving skills in a way in which they are unfamiliar [6-

[7]. In these introductory programming classes, students also have to acquire the syntax, vocabulary and punctuation of the programming language, whether it be C, Java or MATLAB, all of which are popular choices at colleges and universities [6-11].

While first-year, college students may find learning a programming language to be a challenge; there is no question that this knowledge is growing increasingly important in the work environment [12]. Business leaders, educators and politicians are all concerned with developing a 21st century workforce that is prepared to deal with the challenges of a technologically sophisticated environment [12-13]. Now, and in the future, leaders want programming languages taught to all students beginning in elementary school and progressing throughout college [12-14].

Students today want to acquire skills that will help them succeed in the workplace and leaders recognize that knowing and being able to use a programming language is a part of that skill set. Educators, however, have a slightly different focus. Whether it be at a college level or lower, teachers want to be effective in transmitting knowledge to their students in a manner that facilitates learning and skill development and that students find motivating. This is certainly true when teaching a programming language and is the primary focus of the present research project [5, 7].

Learning a programming language is similar to learning a foreign language, as many educators and researchers now realize [8, 12-17]. Both have a vocabulary, syntax and grammar, and some even have their own alphabet [8, 15, 17]. If these two types of languages have similarities, then it is only logical to ask if best practices applied to foreign language teaching could also be applied to teaching a programming language [8, 15, 17].

The presented study examined whether second language acquisition (SLA) teaching practices applied to teaching a programming language would facilitate student learning. The project, which was begun in 2015 and will conclude in May 2017, identified SLA best practices and then translated these into the college classroom in order to provide programming language instruction. New materials were developed and instructors were trained in SLA-based teaching techniques. When acquiring a foreign language, learners progress through levels of proficiency from pre-production to advanced fluency and different teaching techniques work best at each level. This knowledge was translated and then applied to teaching a college-level, introductory computer programming class. Table 1 summarizes the levels of proficiency and then presents teaching strategies applied at each level [18].

Table 1

A Comparison of Non-SLA-based and SLA-based Teaching Techniques

	Preproduction (minimal compre-hension)	**Early Production (limited compre-hension)**	**Speech Emergence (increased compre-hension)**	**Intermediate Fluency (very good compre-hension)**	**Advanced Fluency**
Non-SLA Based Strategies	Few pictures, topics not well explained. Limited self-testing about questions in screencasts.	Multiple choice questions but no simple programs. Facebook used, but no group discussion.	Students begin reading/writing programs to solve engineering problems.	Students given challenging problems to synthetize learning.	Open-ended engineering project used to challenge understand-ing and expand knowledge.
Teaching Strategies in SLA-aBLe	Use pictures and visuals; speak slowly, use simple, shorter words to draw connection between SLA and programming languages; Reinforce learning by giving more self- testing questions.	Ask students to produce simple programs in addition to multiple-choice questions; use group discussion via Facebook.	Emphasize tiered questions and use “think, pair, share” to process the new concepts.	Emphasize compare and contrast for different concepts. Allow students to explain their problem solving process.	Project presentations by students to enhance learning through Q&A.
Specific SLA-based in-class exercises	Show me… Circle the… Where is the…	Yes/No questions Either/Or questions Use 1-2 word answers Use lists and labels	Ask why and how questions Ask students to explain using phrase or short sentence answers	Use ‘What would happen if…’ questions Use ‘Why do you think’ questions	Use ‘decide if‘ exercises Have student ‘retell’ in his/her own words

Table adopted from Frederick et al. (2016).

Problem Statement

Many college students, even in engineering and the other STEM disciplines, find learning a programming language difficult [2, 5]. It is, however, a skill that is needed in today's technologically rich work environments [12, 13]. Therefore, it is important to identity practices that will help students learn a programming language. The present project tested the hypothesis that use of SLA techniques will be effective in teaching a programming language and will result in higher levels of engagement and motivation in students being taught with SLA techniques, as well as better performance in the class. This hypothesis was tested by comparing students in SLA-aBLe versus non-SLA-aBLe introductory computer programming language classes occurring across three semesters at a technological University.

The Current Project

The results presented in the current paper are the culmination of 3 semesters of work on the SLA-aBLe project funded under the RIGEE Program at NSF. The project is described as follows: "The current project applied second language acquisition techniques to teaching an introductory programming language using MATLAB. The project was titled SLA-aBLe, which refers to the use of a SLA approach within a blend learning (aBLe) environment. Three instructors taught EGR 115, an Introduction to Programming Language course using both SLA and non-SLA materials. Each instructor had one section of each class type, with one instructor teaching two non-SLA format classes. In order to help control for instructor differences in teaching, all instructors were trained in the SLA strategies, used the same videos, coordinated their syllabi to cover the same topics and attended regular team meetings to calibrate progress. The SLA sections used 25 innovative, self-paced videos to facilitate student learning for five topics, as well as integrating techniques into classroom teaching that have been shown to be effective in second language acquisition. These cognitive techniques included focusing on a continuum of learning from preproduction to advanced fluency (see Table 1 above). As students progressed across the continuum, they were exposed to materials in different ways specific to their fluency level. In the pre-production phase, for example, learning occurred using visual representations and moderated online discussions. Special videos helped build stage one and two fluency. The videos focused on five important topics: introduction to MATLAB, data types, input and output, conditional statements, and loops. Each video provided definitions, examples and quiz questions to reinforce correct learning. The videos were self-paced so that students could view them as many times as they wished until comprehension occurred. An online, mediated discussion helped support early production skills. At the intermediate level, a 'think, pair, share' technique was used during labs. Intermediate fluency was facilitated through homework, and advanced fluency was achieved by an open-ended project at the end of the semester. To facilitate learning at the intermediate and advanced levels, students were given guided exercises during labs that they then finished on their own. The course culminated in an individual project chosen by the student that used knowledge gained throughout the semester.

Students had the chance to present their projects to the class to show their competence and level of comprehension of the material. Students in the non-SLA sections of the course also used the blended learning environment, but they did not have access to the SLA-aBLe videos, nor did the instructors use SLA-based teaching techniques in those sections." (page 3, Frederick et al., (2016), see reference #18)

Method

This paper presents the results of the SLA-aBLe project from three semesters of implementation. Across the three completed semesters of the project, 267 students participated in 10 SLA-aBLe sections of EGR 115 Introduction to Computing for Engineers. Two hundred, ninety-two students participated in 11 non-SLA-aBLe sections of the same class. This paper will focus on three elements of the project. First, were there differences in perceived motivation and workload between students in SLA-aBLe and non-SLA-aBLe sections of the class? Second, were there differences in grades between students in SLA-aBLe and non-SLA-aBLe sections of the class? Third, did students perceive differences in instruction in SLA-aBLe versus non-SLA-aBLe sections of class, as assessed by end of course evaluations? Measures used in this study included the Intrinsic Motivation Inventory[19], which assessed student motivation across five dimensions, interest/enjoyment, perceived competence, effort, felt pressure and tension, and perceived choice, and the NASA TLX, a well-established measure of self-assessed workload. The IMI was validated for use with college student populations [20]. The NASA TLX was validated by researchers at NASA[20]. The TLX measures six workload dimensions: mental demand, physical demand, temporal demand, performance, effort and frustration [20]. The IMI and TLX were each administered six times across all semesters studied (at beginning and end of course, and after each of the four learning videos). Grades for each EGR 115 section were collected at the end of each semester. The present paper examines the following research questions:

1. Were there differences in perceived motivation and workload between students in SLA-aBLe and non-SLA-aBLe sections of the class? As was done after the first year of the project, this question was assessed using *t*-tests with section type as the independent variable and the IMI and TLX variables entered as dependent variables. For all tests, the significance level was set at p=.05. For the motivation variables, we predicted that students in the SLA-aBLe sections of the class would report higher levels of interest/enjoyment, competence and effort, and lower levels of pressure/tension. No prediction was made about perceived choice. For the TLX, we predicted students in the SLA-aBLe sections of the class would report lower levels of frustration, with no prediction made about the differences on the other workload variables.

2. Were there differences in grades between students in the SLA-aBLe and non-SLA-aBLe sections of the class? If SLA-based instruction is effective, students in those classes should show better performance in the class than their peers in non-SLA-based sections of the class. This analysis was conducted using a chi-square analysis with significance level set at p=.05.

3. Did students perceive differences in instruction in SLA-aBLe versus non-SLA-aBLe sections of class, as assessed by end of course evaluations? To address this question, mean instructor ratings on four end of course outcomes are presented: course clarity, course content and organization, achievement of overall learning outcomes and ratings of student-instructor interaction.

Results

Question 1: Were there differences in perceived motivation and workload between students in SLA-aBLe and non-SLA-aBLe sections of the class?

Differences in Perceived Workload. Differences in perceived workload between students in SLA and Non-SLA sections occurred only during the first week of the semester and at the end of the class. During the Week 1 survey administration, SLA-aBLe students reported significantly higher levels of perceived physical and temporal demand and significantly lower levels of effort than their fellow students in non-SLA-aBLe sections of the class. At the end of the course, students in the non-SLA-aBLe sections of the course reported a significantly higher level of perceived effort than students in the SLA sections of the class. During other survey administration periods, no statistically significant mean differences between groups were found. These results are presented in Table 2 below.

Differences in Motivation. Significant motivational differences were found between students in SLA-aBLe course sections and students in non-SLA-aBLe sections twice across the administration periods. After viewing the data types' video and the loops video, students in the SLA-aBLe course sections reported significantly higher levels of competence than students in the non-SLA-aBLe sections of the class. No other motivational comparisons reached statistical significance. These results are also presented in Table 2 below.

Table 2
Means for Workload and Motivation Variables across Administration Periods for 3 Semesters

		Administration Period					
		Week 1 of Course	**Data Types**	**Input / Output**	**Conditional Statements**	**Loops**	**End of Course**
Workload Variables	Class Section	Means of NASA TLX					
Mental Demand	SLA (n=86) Non-SLA (n=80)	9.58 8.44	11.11 12.42	12.28 12.81	11.81 12.41	13.43 13.14	15.65 16.71
Physical Demand	SLA (n=86) Non-SLA (n=80)	4.90 3.89**	5.72 5.40	7.40 5.38	6.08 5.00	6.98 6.07	6.90 7.16
Temporal Demand	SLA (n=86) Non-SLA (n=80)	9.46 7.17**	11.04 10.88	11.85 11.26	11.85 11.16	11.39 12.41	15.10 16.13
Performance Demands	SLA (n=86) Non-SLA (n=80)	6.18 6.07	6.69 8.04	8.53 6.26	7.88 7.73	7.95 8.03	7.23 9.03
Effort	SLA (n=86) Non-SLA (n=80)	11.57 9.68**	12.15 12.65	13.34 14.15	13.35 12.78	13.07 13.34	15.55 17.13**
Frustration	SLA (n=86) Non-SLA (n=80)	7.74 7.16	9.22 10.25	10.02 10.43	8.94 10.95	10.77 11.07	13.50 14.81
Motivation Variables	IMI						
Enjoyment	SLA (n=86) Non-SLA (n=80)	4.44 4.30	4.21 4.27	4.74 4.14	4.28 4.07	4.27 4.00	4.16 4.39
Importance	SLA (n=86) Non-SLA (n=80)	4.95 4.59	5.02 5.03	5.64 5.13	5.20 5.14	5.15 5.18	5.70 6.04
Pressure-Tension	SLA (n=86) Non-SLA (n=80)	2.84 2.52	2.98 3.27	3.21 3.77	2.99 3.47	3.33 3.52	4.30 4.54
Competence	SLA (n=86) Non-SLA (n=80)	4.75 4.96	5.14 4.55*	5.29 4.88	5.10 4.82	4.80 4.51**	4.53 4.24
Usefulness	SLA (n=86) Non-SLA (n=80)	4.94 5.18	5.23 5.04	5.68 5.10	5.36 5.03	5.17 5.04	4.89 4.97

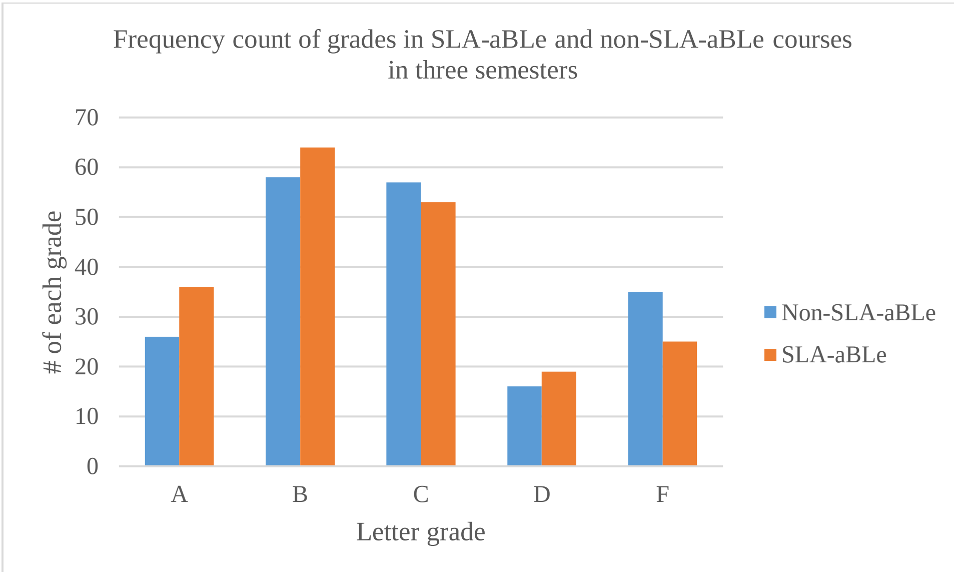
* *p*<05
** *p*<.01

Question 2: Were there differences in grades between students in the SLA-aBLe and non-SLA-aBLe sections of the class?

A chi-square test of independence showed no statistically-significant relationship between the course type and final grade. Students in the SLA-aBLe sections did not achieve significantly higher grades in the class than students in the non-SLA-aBLe sections, even though the percentage of ‘A’ and ‘B’ grades was higher in the SLA-aBLe sections, while the percentage of ‘F’ grades was lower. Table 3 presents a chart showing this information.

Table 3

Comparison of students' final grades in the SLA-aBLe and non-SLA-aBLe sections for three semesters – Fall 2016, Spring 2016, Fall 2016



Question 3: Did students perceive differences in instruction in SLA-aBLe versus non-SLA-aBLe sections of class, as assessed by end of course evaluations?

Overall instructor ratings are presented in Table 4 below. These items are composite items that are included on all end of course evaluations. There are four overall instructor ratings based on student perceptions: course clarity, course content, learning outcomes and student/instructor interaction. Scores on all of these items range from 1 to 4, with 1 being the lowest rating (strongly disagree) and 4 being the highest rating (strongly agree). No statistical testing was done for this research question, because at the present time data collection are not yet complete for the second year of the study. From examination of the ratings, it appears that differences may exist between student ratings from Year 1 to Year 2 of the project, more so than between SLA-aBLe and non SLA-aBLe sections of the class, however no firm conclusions can be drawn until Year 2 of the project is completed.

Table 4

Mean Instructor Rating on Overall Course Outcomes: A Comparison Between Year 1 and Year 2 of the Project

Evaluation Item	SLA-aBLe Students Year 1+	SLA-aBLe Students Year 2*	Non SLA-aBLe Students Year 1+	Non SLA-aBLe Students Year 2*
	Mean (number of sections assessed)		Mean (number of sections assessed)	
Overall Clarity of Presentation	3.42 (5)	3.36 (4)	3.47 (8)	3.43 (2)
Overall Content, Structure and Organization of Class	3.26 (5)	3.23 (4)	3.30 (8)	3.24 (2)
Overall Learning Outcomes were Achieved	3.44 (5)	3.33 (4)	3.46 (8)	3.24 (2)
Overall Student/Instructor Interaction was Positive	3.55 (5)	3.59 (4)	3.60 (8)	3.57 (2)

+ Year 1 data includes 2 of 3 instructors

* Year 2 thus far contains only Fall 2016 ratings

Discussion

The SLA-aBLe Project is a work in progress. Instruction using SLA techniques began in fall 2015 and has now spanned three semesters with a fourth semester currently ongoing. The project concludes at the end of the current semester and final results of the study will be compiled beginning in May 2017. This project integrated SLA teaching techniques into an introductory programming course, and then compared course perceptions and outcomes in the SLA-based classrooms to the same perceptions and outcomes in Non-SLA-based classrooms of the same course. The goal of the project is to determine if SLA-based instruction can facilitate student learning for a computer programming language.

This WIP paper presented data collected across the first three semesters of project implementation. Outcomes related to student perceptions of workload and motivation were presented, as well as grades in the class, and ratings on end-of-course evaluations. Some caution should be used in examining the results from the project, since data collection and analyses are not yet complete.

The first question the paper addressed was whether or not workload and motivational differences existed between students in SLA-aBLe and non-SLA-aBLe sections of the course. Results from this analysis were modest. In terms of workload, during the first week of the class, students in SLA-aBLe sections of the class perceived several workload variables to be heavier than students in the Non-SLA sections. However, by the end of the semester, differences in workload became insignificant, except for a difference in effort showing that non-SLA-aBLe students felt they put more effort into the class. Each semester, students in all sections of the class (SLA and non-SLA) are briefed on the project and told they may be in a section that will be using new learning materials, and that they will be asked to answer questions about their experience throughout the semester. It may be that, as a result of that briefing, the students in the SLA-aBLe sections felt their workload might be heavier, although over the course of the semester that perception changed.

There were few significant motivational differences across students in SLA-aBLe and non-SLA-aBLe sections of course. The one significant difference that was observed was for competence. Students in the SLA-aBLe sections of the class reported higher perceptions of competence after learning material related to data types and loops, than students in the non-SLA-aBLe sections. This difference may reflect positively on the specialized videos that were developed for the project, and which students in the SLA-aBLe sections watched during those weeks.

The second research question compared final course grades between students in SLA-aBLe and non-SLA-aBLe sections. No statistically-significant differences occurred in course grades, even though a higher proportion of students in SLA-aBLe sections received an ‘A’ or ‘B’ grade and a lower proportion received an ‘F’ grade, than students in the non-SLA-aBLe sections. This trend is promising. Regardless of the teaching technique used in a section, the department is responsible for the standardized material covered, as well as some of the testing materials. Instructors maintain some flexibility in how grades in the course are calculated, but those instructors teaching both SLA-aBLe and non-SLA-aBLe sections did not vary how they calculated final grades between the two types of sections.

The last research question asked was: Do outcomes on end-of-course evaluations vary for SLA-aBLe and non-SLA-aBLe sections of the class? At the current time, results would indicate that few, if any, end of course differences occurred between SLA-aBLe and non-SLA-aBLe sections. However, the pattern of results does seem to point toward evaluation differences between Year 1 and Year 2 of the project. Year two data collection is only halfway complete, so any firm conclusions about differences in how students evaluate the course cannot be made until the end of the current semester.

In today's technology-rich work environments, learning a programming language is becoming a necessary skill. However, many students still find programming languages hard to learn. Instructional materials and techniques need to be developed to help students of all ages acquire that important skill. With that in mind, any materials and techniques that are developed should be tested and analyzed for effectiveness, as is being done in the current project.

In summary, the current paper presented results from two years (3 semesters) of an NSF funded research project that integrated SLA techniques into an Introduction to Programming language class and then compared outcomes for an SLA-based class to outcomes for the same class taught in a non-SLA based manner. The project's goal is to determine if an SLA-based approach to instruction can facilitate learning of a programming language.

Acknowledgements: The authors are grateful for the support provided by the National Science Foundation, Division of Engineering Education and Centers, grant number EEC 1441825. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors also would like to acknowledge the effort from Ms. Caroline Liron, Dr. Matthew Verleger, who helped conduct the project in their classes, and Michael Chronister and Lauren Massey from the Institution Research Department at Embry-Riddle Aeronautical University who collected and helped analyze the survey data for this project. In addition, Dr. Frederick would like to thank the Department of Human Factors at Embry-Riddle Aeronautical University for supporting travel and other activities related to the project.

Bibliography

1. Bualuan, R. (2006). Teaching Computer Programming Skills to First-year Engineering Students Using Fun Animation in MATLAB. Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL.
2. Burton, P.J. & Bruhn, R.E. (2003). Teaching Programming in the OOP Era. ACM SIGCSE Bulletin, Volume 35, No. 2, pages 111-114.
3. Devnes, P.E. (1999). MATLAB and Freshman Engineering. Paper presented at the 1999 American Society for Engineering Education Annual Conference & Exposition, Charlotte, NC.
4. Morrell, D. (2007). Design of an Introductory MATLAB Course for Freshman Engineering Students. Paper presented at the 2007 American Society of Engineering Education Annual Conference & Exposition, Honolulu, HI.
5. Naraghi, M.H.N. & Litkouhi, B. (2001). An Effective Approach for Teaching Computer Programming to Freshman Engineering Students. Paper presented at the 2001 American Society for Engineering Education Annual Conference & Exposition, New York.
6. Partovi, H. (2013). Computer Programming Education Needed, USA Today, February 26 2013. Retrieved from http://www.usatoday.com/story/opinion/2013/02/26/computer-programming-coding-education/1947551/.
7. Victor, B. (2012). Learnable Programming. Retrieved March, 7, 2014, from http://worrydream.com/LearnableProgramming
8. Solomon, J. (2004). Programming as a Second Language. Learning & Leading with Technology, Volume 39, No. 4, 34-39.
9. Azemi, A., Pauley, L.L. (2006). Teaching the Introductory Computer-Programming Course for Engineering Using MATLAB and Some Exposure to C. Paper presented at the 2006 American Society for Engineering Education Annual Conference & Exposition, Chicago, IL.

10. Yale, M., Bennett, D., Brown, C., Zhu, G., & Lu, Y. (2009). Hybrid Content Delivery and Learning Styles in a Computer Programming Course. Paper presented at the 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, TX.
11. Sun, L., Kindy, M., & Liron, C. (2012). Hybrid Course Design: Leading a New Direction in Learning Programming Languages. Paper presented at the 2013 American Society of Engineering Education Annual Conference & Exposition, San Antonio, TX.
12. Clark, K.M. (2017). Spanish, French and SQL? The Push to Teach Coding Like a Foreign Language. Tampa Bay Times, February 7, 2017.
13. Tran, L. (2014) Computer Programming Could Soon be Considered a Foreign Language in One State. Retrieved March 7, 2014, from http://www.policymic.com/articles/81067/computer-programming-could-soon-be-considered-a-foreign-language-in-one-state
14. Tyre, P. (2013) Is Coding the New Second Language? Retrieved March 7, 2014, from http://www.smithsonianmag.com/innovation/is-coding-the-new-second-language-81708064/
15. Van Roy, P., (2003). The Role of Language Paradigms in Teaching Programming. Paper presented at the 34th SIGCSE Technical Symposium on Computer Science Education, New York, NY.
16. Wynn, M., (2015). Ky. Ponders Teaching Computer Code as Foreign Language. Retrieved January 29, 2015, from http://www.usatoday.com/story/tech/2015/01/29/ky-computer-code-as-foreign-language/22529629/
17. Sun, L., Frederick, C., & Liron, C. (2015). Applying Second Language Acquisition to Facilitate Blended Learning of Programming Languages. Proceedings of the ASEE Annual Conference and Exposition, Seattle, WA.
18. Frederick, C., Sun, L., Liron, C., Verleger, M., Cunningham, R. & Sanjuan Espejo, P. (2016). Implementation and evaluation of a second language acquisition-based programming course. Proceedings of the ASEE Annual Conference and Exposition, New Orleans, LA, June 28, 2016.
19. McAuley, E., Duncan, T., & Tammen, V. V. (1987). Psychometric Properties of the Intrinsic Motivation Inventory in a Competitive Sport Setting: A Confirmatory Factor Analysis. Research Quarterly for Exercise and Sport, 60, 48-58.
20. Hart, S. G. & Staveland, L. E. (1988) Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock and N. Meshkati (Eds.) Human Mental Workload. Amsterdam: North Holland Press.