INTRODUCTORY COMPUTER PROGRAMMING COURSE TEACHING IMPROVEMENT USING IMMERSION LANGUAGE, EXTREME PROGRAMMING, AND EDUCATION THEORIES

by

Miguel Velez-Rubio

BERNARD J. SHARUM, PhD, Faculty Mentor and Chair

CLIFFORD BUTLER, PhD, Committee Member

LINDA K. DELL'OSSO, PhD, Committee Member

Sue Talley, EdD, Dean, School of Business and Technology

A Dissertation Presented in Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

Capella University

July 2013

Abstract

Teaching computer programming to freshmen students in Computer Sciences and other Information Technology areas has been identified as a complex activity. Different approaches have been studied looking for the best one that could help to improve this teaching process. A proposed approach was implemented which is based in the language immersion theory combined with extreme programming techniques and strategies associated with educational concepts, including collaborative learning, problem based learning, self-directed learning, and experiential learning. A study using the mixed method was worked. Combining qualitative and quantitative results through the triangulation approach, two groups were defined for an experiment that compared a group of students working with the traditional approach for teaching computer programming with another group using the proposed approach. The students' performance reflected on the posttest scores determined the main dependent variable evaluated from both groups. The groups constitute the independent variable. Throughout classes qualitative information was compiled using the Critical Incident Questionnaire for formative assessment. A questionnaire was administered to evaluate students' motivation toward programming, and another one to evaluate students' predisposition to continue in the program. In the experimental group the last activity was a focus group to evaluate the opinion of students regarding the proposed approach. The results demonstrated that the proposed approach was not significantly different from the traditional approach in terms of its efficiency. Furthermore, the data collected demonstrated that the proposed approach was not significantly better than the traditional one. Student's comments revealed they felt comfortable working with the proposed approach. This supports the

results demonstrating it was efficient; however, this is not enough to establish that it is better than the traditional approach. Limitations were identified and considered instrumental in the results obtained. It is important to make adjustments to the design used and to replicate this study in diverse contexts in order to confirm the results obtained or to demonstrate what is still believed to be a valuable teaching approach.

Dedication

I dedicate this work to my wife, Marta, my son Ian Miguel, and my new born daughter Isis Mariel for their patience and support, and to my parents Miguel and Altagracia, for being there at all times. Thanks God for your support!

Acknowledgments

I must acknowledge the people that believed in me, and supported me throughout this endeavor. To those who supported me in different ways in my job from the beginning to now: Dr. Héctor De Jesús Cardona, Prof. Irma Schmidt, Dr. Arturo Avilés, Dr. Edna Miranda, Dr. Javier Ávalos, and Dr. Orlando González. To those special people who gave me additional support in important situations: Prof. Antonio F. Huertas, Dr. Maritza Sostre, and Dr. Jorge Rovira. To an excellent person from Capella University that made more than required in supporting me: Dr. Bernard J. Sharum, my mentor. To my motivators, supporters and the most engaged in this process: Marta, Ian Miguel, and Isis Mariel, and my parents, Miguel and Altagracia.

Table of Contents

List of Tables

List of Figures

CHAPTER 1. INTRODUCTION

Introduction to the Problem

The teaching of computer programming to freshmen students in computer sciences and other related areas has been identified as a very complex task. The literature reveals that for some students the learning of programming skills is a real challenge. For them, acquiring the knowledge and skills associated with programming computers is very difficult. Sheard, Simon, Hamilton, and Lönnberg (2009) presented an excellent survey study in which this situation was extensively discussed in light of the different research papers they analyzed. Through that work, it was demonstrated that a lot of effort were expended addressing the challenging situation of teaching computer programming. Diverse additional studies also addressed this problem extensively in different instructional contexts used for teaching computer programming as occurred in Bennedsen and Caspersen (2008), Blanco et al. (2009), Daly (2011), Esteves, Fonseca, Morgado, and Martins (2011), Hawi (2010), Pokorny (2009), Thomas, Ge, and Greene (2011), and White (2006).

Computer programming professors constantly assess the approach used in teaching programming, making adjustments in order to look for improvements in their classes' results. In the process, they adopt ideas from researchers that they considered appropriate. An example of those ideas is the use of robotic characters that accept simple programming commands as instruments for teaching computer programming skills. Sprankle (2005) referred to an imaginary robot named Otto, of her creation, used to help in teaching initial algorithm development concepts. Another faculty member was using Karel the robot (Patfis, 1981; Becker, 2001; and Borge, Fjuk, and Groven, 2004). Karel was implemented in software with different versions and languages even allowing students to write real programming code to manipulate it.

A new teaching approach is proposed based on the integration of some ideas already in use in teaching computer programming classes with techniques obtained from immersion language theory, extreme programming, and some education theories. It is expected that, by using this modified approach to teach programming to first year computer sciences students, an improvement can be identified in the final results obtained after evaluating the knowledge and skills acquired. Students' experience can be addressed in terms of their motivation toward programming and their predisposition to continue in the program.

Background of the Study

The area of Information Technology (IT) Education is receiving a lot of attention from researchers in recent years. There is a significantly high amount of articles published addressing different aspects associated with this area. A good evidence of this is the work presented by Sheard et al. (2009) in which, as mentioned, they evaluated a substantial amount of articles published only in a specific area of IT education; the teaching and learning of computer programming.

In the world of research, the teaching of computer programming means covering diverse issues in a variety of areas. However, within the context of IT education for those beginning in the IT world, it is very important to be assured of teaching programming logic and programming languages effectively. This is the only way to assure that future professionals in this area are adequately prepared because the foundations are the most important part of the process.

Most articles address the specific issues associated with the assessment of appropriate techniques to teach programming logic to first year IT students. Others search for the best practices in teaching programming languages. In first year courses, teachers work to develop the programming logic skills needed by students in producing good algorithmic solutions. In reference to first year students Pokorny (2009) established that "[m]any of those entering the first course are unprepared for computer programming" (p. 166). This shows the need to explore the best techniques for teaching programming effectively.

Teaching Programming Logic Methodology and the Associated Context

The literature available addressing issues related to teaching programming logic is diverse in the context of the experimentation developed. The relevance of the strategies used to teach programming logic or programming languages is independent of the context in which some works were developed. There may be techniques in use by teachers in a graduate programming course that can be useful in undergraduate courses. This can occur also in teaching programming concepts to younger people. However, some techniques may be appropriate with some groups but will not necessarily work with others.

The methodologies and techniques found in the literature are diverse. Sheard et al. (2009) established that "programming is hard to learn and therefore hard to teach well, so there is a great deal of literature around it" (p. 94). The study they presented covers the analysis of numerous research papers in teaching programming. They analyzed the research approach used, the validity and reliability of the articles and areas for further investigation. This work is very important in IT education. It provides a good idea of the status of research in the field.

An aspect that can be viewed through the different studies referred by Sheard et al. (2009) is the importance of the context in teaching programming. The context can be associated with a variety of things. Some of them are the teaching methodology established by the professor, the modality used for the course (onsite, online or blended), and the cultural context in which the course is offered.

When exploring the teaching of computer programming in different contexts, the work of Pokorny (2009) is one of importance. Pokorny presented a softer approach to teach first year students in computer disciplines, the fundamental concepts of the area, including programming. About this the author wrote, "[t]he new approach is intended to provide a softer introduction to computer programming and a stronger introduction to the various disciplines within computing" (p. 168). The approach was focused on problem solving and the role of computers to solve problems with algorithmic solutions.

Blanco et al. (2009) established an approach that was opposed to Pokorny's (2009) idea. Blanco et al. proposed a methodology based on formal specification and program calculation to teach programming logic to Computer Sciences students. The approach involved the teaching programming from the beginning using functional programming instead of procedural structured programming or object oriented programming. They used three methods to collect data: students' produced material evaluation, personal interviews and in class observations. This study has two aspects that may be significantly important in producing uncertain results. One was the development of a methodology that is evidently more complicated than the methodologies normally used in teaching programming. The other aspect is about the spoken language used in teaching that is Spanish, as it occurs in Puerto Rico. How much complicated was the situation when considering they were working with the three factors combined? It may be interesting to know.

The study presented by DePasquale (2003) was about the "lack of quantitative results examining the effect of" programming environments upon freshmen students of Computer Science. The purpose was to examine the effects of these environments examining the effects of a proposed prototype environment. The formulated hypothesis was "novice students do not need such complex tools and that the introduction of a highly complex structured programming environment for these students is detrimental to a clear understanding of their first programming language and programming in general" (p. 21). The results of the study were in general positive in favor of the hypothesis presented but with moderate consistency in terms of the significance of the results. Again the context is an important aspect to be considered.

The study presented in Govender (2009) addressed the importance of the context. According to Govender, the learning context includes the lectures, the study process, the previous knowledge or teaching experience, and the tests. The learning context was argued to have important implications in the teaching and learning of programming. The study was intended to understand this phenomena using activity theory for the analysis. Gevender found that both the problem and the learning context have an important effect on students' performance and understanding. In the context area also Hawi (2010) found in a study with computer programming students that ten different causal attributions were the responsible for the success or failure of teaching process. Some of the attributions were *lack of practice*, *subject difficulty*, *appropriate teaching method*, *lack of time*, and *unfair treatment*.

The referred studies are part of what is found in the literature addressing a variety of issues and situations pertaining to the learning context of computer programming. The literature review presents and discusses further studies.

Statement of the Problem

Teaching programming to computer science freshmen students in general is recognized as a challenging situation resulting in students leaving the academic computing programs, as is documented in the literature. Most research was accomplished exploring different approaches that facilitate this process and that allow teachers to be efficient in providing students with the knowledge and skills required (Sheard et al., 2009). Most of the ideas proposed are reasonably useful in researching processes, though some are not. However, the most efficient approach for teaching programming to first year students may have not been found yet, so there is a lot of work to be done in this area.

Purpose of the Study

The purpose of this study is to examine if the implementation of a new approach for teaching computer programming to freshmen students in computer sciences causes an improvement in the acquisition of programming knowledge and skills, in comparison to students that receive the classes using the established traditional approach. It is expected that this can result in a faster acquisition of the needed skills and in producing more confidence on students in programming code development.

Rationale

The new teaching approach is expected to demonstrate an improvement in students' performance at the end of the initial course. The improvement in efficiency obtained in solving computing problems through programming is expected to result in motivation for students to continue in the computer science program. The traditional approach is based on an incremental approach beginning with the simplest ideas and ending with the more complex ones while the proposed one follows a different strategy using some ideas from the immersion language theory, extreme programming, and some educational theories.

From the immersion language theory the main idea is associated with developing an experience of total immersion with the computing programming language, even from the initial topics of the course, resolving problems that may be considered adequate for the experience. In Harper (2006) this idea was suggested and tested in a real setup. From extreme programming (Hoffer, George, and Valacich, 2011) some techniques are part of the ideas that can be effectively implemented as part of the new approach that in turn will support the implementation of the language immersion use mentioned. The ideas from both areas combined can be associated with important theories and concepts of education like andragogy, self-directed learning, collaborative learning, and experiential learning, among others (Knowles, Holton, and Swanson, 2011; Merriam, Caffarella & Baumgartner, 2007; McKeachie and Svinicki, 2006; Palloff and Pratt, 2007; Wlodkowski, 2008).

Research Questions

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

The question is associated directly with the idea of determining if the proposed approach resulted in being appropriate for teaching computer programming. It addressed the main problematic situation expressed in the problem statement by providing with an alternative methodology for this educational process. The real efficiency of the proposed approach can be determined by analyzing the specific aspects addressed in the four sub-questions formulated. This question was analyzed using conveniently the results of both the quantitative and the qualitative approach according to the guidance established by Tashakkori and Creswell (2007) about research questions for mixed method studies.

ResQ 1.1. What is the difference between students working with the proposed approach in terms of performance as compared to the performance of the students that received instruction using the traditional approach? (Quantitative Approach)

The question is associated directly with the traditional assessment used in classes based on the scores obtained in the pretest and posttest for both groups. It is the most important measure of effectiveness of the proposed approach.

ResQ 1.2. What is the difference between the two groups of students on their motivation toward programming? (Quantitative Approach)

The motivation of students toward programming is an important measure that can be used in determining the efficiency of the resulting learning process. In order to determine this measure, this researcher used the Modified Rosenberg Self-Esteem Scale as applied by Bergin and Reilly (2006), and Kranch (2010) in studies worked in similar contexts.

ResQ 1.3. What is the difference between the two groups of students in terms of their predisposition toward continuing in the program? (Quantitative Approach)

The predisposition of students to continue in a program is a measure associated with the success in the use of the innovative approach because it may demonstrates students' understanding of programming concepts and what they can expect in the following courses. This was confirmed with the final students' scores and their motivation toward programming. The Ramalingham-Wiedenbeck Computer Self-Efficacy Scale instrument was the one used in determining this measure. It was also successfully used in Askar and Davenport (2009), and Kranch (2010).

ResQ 1.4. How do the students feel in working with the proposed approach? (Qualitative Approach)

It is important to know how comfortable the students were in working with the proposed approach. The experience lived in this context throughout the course was captured using interviews or focus groups. The protocol for the focus group is available in the Appendix.

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

More than knowing if the proposed approach is efficient for teaching programming, this research set out to investigate if the implementation of a new approach causes an improvement in the acquisition of programming knowledge and skills in comparison to students that receive instruction using the traditional approach. In order to determine this, some factors were analyzed that are reflected in the following sub-questions. It was expected that the proposed approach could serve as an effective mechanism for teaching computer programming, allowing students to feel comfortable in their computer programs and allowing teachers and program managers to be more effective in doing their work in classrooms and in course adjustments. This question was analyzed using the results of both the quantitative and the qualitative approach according to guidance in Tashakkori and Creswell (2007) regarding research questions for mixed method studies.

ResQ 2.1. What is the improvement, if any, in the average time needed by students to get confident in programming using the main building blocks and structures required to be learned in the first programming class? (Quantitative Approach)

One of the main aspects of the proposed approach is associated with the belief that it can provide for a shortening in the time needed to understand some of the main concepts and skills associated with programming. If some concepts are learned earlier it can be easier to learn others and the skills associated can be also improved.

ResQ 2.2. Based on the Critical Incident Questionnaire (CIQ) instrument for formative assessment, how do the computer science freshmen students participating in the study benefit from using the proposed approach? (Qualitative Approach)

The CIQ was used periodically in order to get an idea of the real status of all students. It was expected that very accurate information would be obtained using this instrument since it is submitted anonymously and does not have any effect on course's scores. This researcher analyzed the conclusions obtained with the CIQ in order to use their results as part of the statistical analysis that helped to answer the main research questions.

ResQ 2.3. How much time is needed to get confident with the different building blocks and structures while the understanding acquired according to the CIQ is improved using the proposed approach? (Mixed Approach)

The comparison and analysis of the times obtained needed for completing each topic with the results obtained from the CIQ were very important in concluding about the advantage of using the proposed approach over the traditional one. It was expected to find a reduction in the times needed by students to get confident with the different building blocks and structures (represented by different topics in the course) and having expressions in the CIQ establishing students' understanding of them.

ResQ 2.4. How significant is the difference between the two groups of students in performance and to what extent is it in favor of the proposed approach? (Quantitative Approach)

This one is based on question ResQ 1.1.

ResQ 2.5. How significant is the difference between the two groups of students in motivation toward programming and to what extent is motivation a factor in favor of the proposed approach? (Quantitative Approach)

This one is based in question ResQ 1.2.

ResQ 2.6. How significant is the difference between the two groups of students in the predisposition toward continuing in the program and to what extent is it in favor of the proposed approach? (Quantitative Approach)

This one is based on question ResQ 1.3.

Hypotheses

$H_0 1.1$: The means of the scores for the two groups are not significantly different.

$H_A 1.1$: The means of the scores for the two groups are significantly different.

$H_0 1.2$: Students using the proposed methodology were similarly motivated or less motivated toward using programming in comparison to those using the traditional approach.

$H_A 1.2$: Students using the proposed methodology were more motivated toward using programming in comparison to those using the traditional approach.

$H_0 1.3$: Students using the proposed approach were similarly predisposed or less predisposed to continue in the program in comparison to those using the traditional approach.

$H_A 1.3$: Students using the proposed approach were more predisposed to continue in the program in comparison to those using the traditional approach.

$H_0$2.1: The learning of programming concepts for the students using the proposed approach will be similar to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

H_A2.1: The learning of programming concepts for the students using the proposed approach will be faster compared to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

$H_0$2.2: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly similar or negative about their understanding of the concepts presented.

H_A2.2: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly positive about their understanding of the concepts presented.

$H_0$2.3: The time needed to get confident with the different building blocks and structures will appear to be similar or higher and the understanding acquired as reflected in the CIQ will appear to be similar or lower while students learn with the proposed approach.

H_A2.3: The time needed to get confident with the different building blocks and structures will appear to be reduced and the understanding acquired as reflected in the CIQ will appear to be higher while students learn with the proposed approach.

$H_0$2.4: The means of the test scores for the two groups are not significantly different or the mean for the group working with the proposed approach is significantly lower.

H_A2.4: The mean test scores for the group working with the proposed approach is significantly higher.

Significance of the Study

The intention of improving the teaching of computer programming to freshmen students, has been extensively addressed in the literature for over 40 years, as can be deduced from Bennedsen and Caspersen (2008). Faculty in computer sciences and other IT related programs teaching programming have been constantly involved in the process of improving the approach used in classes. This is in order to provide the needed skills and knowledge as early as possible assuring at the same time a better understanding of the process. A good understanding of the process is associated with an improved motivation for students (motivational theories). Motivated students can do a better job and may exhibit more confidence about their competence to develop programs efficiently (self-efficacy theory). Also there is a better probability that motivated students opt to continue in the program, promoting the improvement of retention rates.

From the implementation and testing of the new approach it is expected that the definition of curricula in computer sciences and other related disciplines, as defined in Association for Computer Machinery (2005), can be significantly improved, especially in the development of programming concepts. After having the possibility of improving the learning experience of students in the initial course, the more advanced courses can be also improved because students may have a better understanding of the programming concepts and the associated skills. In this sense, the contribution of the study can be significantly important if it is found that the proposed approach is effective in strengthening the teaching process.

Contributions to the Field

Throughout the years most researchers related with academic computing programs had worked in developing new approaches to teaching computer programming more effectively. A variety of tools, strategies, and approaches had been studied and tested looking for the best one that could solve a significant problem. It is about improving the retention rates of the different programs. Retention rates are mainly associated with the problem of adequately providing the knowledge and skills related to the programming process. Teaching computer programming to freshmen students turns out to be particularly complex. The main contribution of this study to the field is precisely in providing for approach to teaching computer programming to freshmen students.

Definition of Terms

Extreme programming (XP). It is a programming development approach considered part of the agile software development methodology. It was defined as coding based on customer specifications and testing the code ensuring that prior steps are aligned with the intended development, according to Erickson, Lyytinen, and Siau (2005).

Language immersion. According to Harper (2006) it is a technique consisting in leading students to speak exclusively the language to be learned from the first class.

Pair programming. One of the twelve core practices associated with XP consisting on working programs in couples in order to share ideas about the code making the process more effective and agile. As Goel and Kathuria (2010) established, "[p]air programming is a situation in which two programmers work side-by-side, designing and coding, while working on the same algorithm" (p. 186).

Programming language immersion. This implies to promote an experience of immersion in the context of a programming language by leading students to analyze and interpret statements directly from an already working program in order to determine how the whole program works.

Programming topic immersion. An experience of immersion to be promoted working earlier in a semester with topics that are more complex than those normally taught in that moment, requiring students to work with them.

Assumptions, Strengths and Limitations

Assumptions

Theoretical assumptions

- The extreme programming methodology is generally applied in the context of the formal development of software by experienced programmers. It is assumed that some techniques from extreme programming can be used in the context of teaching computer programming to freshmen students.
- Self-directed learning is an assumption associated with andragogy, a concept established by Knowles as mentioned by Merriam, Caffarella, and Baumgartner (2007) and Knowles, Holton, and Swanson (2011). It will be used in the study as part of the mechanisms needed to implement the proposed approach.
- Problem based learning is assumed to be a teaching approach of common use in teaching computer programming languages. This has an important implication in the idea of having courses that by definition are intensive in providing practice in the solution of problems. Similarly this implies that students are exposed to elements of experiential learning, making of the teaching-learning process a richer one. This is expected to produce an improved experience for those students working with the

proposed approach, assuring that they can really learn something in the process, as McKeachie and Svinicki (2006) mentioned.

Topical assumption

- The immersion language theory is the basis of the study. It is assumed that the theory can be applied to computer languages as proposed by Harper (2006) and in the study.

Methodological assumption

- In analyzing the dependent variable of the study, parametric tests can be used because the samples are independent, they are assumed to be normally distributed and to have equal variances, and the measurement scales are intervals (final scores), according to Cooper and Schindler (2011).

Strengths

One of the main strengths of the study was the automatic random distribution of subjects in the two groups. This automatic random distribution, that produces a randomly like selected sample, was established by the individual decision of students in selecting a class schedule according to their own preferences and the available spaces. Towards the end of the study, the ten participants for the focus group were also randomly selected from the experimental group.

An additional strength of the study is in the fact that it was totally developed in real classroom setups with real freshmen students from an already running computer science program of an accredited institution with recognized long experience in the area. This situation allowed ascertains that the results can be generalized to similar institutional setups around the country. This is mainly true because the general methodology used for teaching computer programming in all institutions is almost the same.

The third strength is in the research methodology defined for the study. A study of this nature may be conducted with fewer elements included. For example, it can be developed using only quantitative measures including the posttest scores and the two questionnaires presented. However, the qualitative elements finally included were helpful in validating and supporting the results obtained that in turn helped in strengthening the limitations existing. This implies an important aspect of any study working with qualitative data as pointed out by Swanson and Holton (2005) that is producing an improvement in the consistency of the findings. The qualitative data also served to consider aspects associated with the opinions and feelings of the participants of this research.

An additional strength is in the conviction of having a proposed approach that will really guide students to a learning experience that is considered better than any other approach previously used for teaching computer programming. This will be beneficial for students not only in the first course but also in all subsequent courses associated with programming.

Limitations

The main limitation of the study is the small sample used. As established the groups were expected to be of 20 participants in each for a total of 40. This is normally considered a small sample for most studies. However, as mentioned, this limitation was in some way minimized by using a mixed method approach because the obtained quantitative results were corroborated with qualitative elements that were helpful in improving the quality of the study.

Another limitation identified is the assumption that participants respond honestly to the items of the Critical Incident Questionnaire (CIQ) and to the questions of the focus group.

A third limitation identified is the researcher's decision of conducting the experiment in his working institution. However, this situation was not identified as one that can cause any potential risk for participants or any bias within the context of the study.

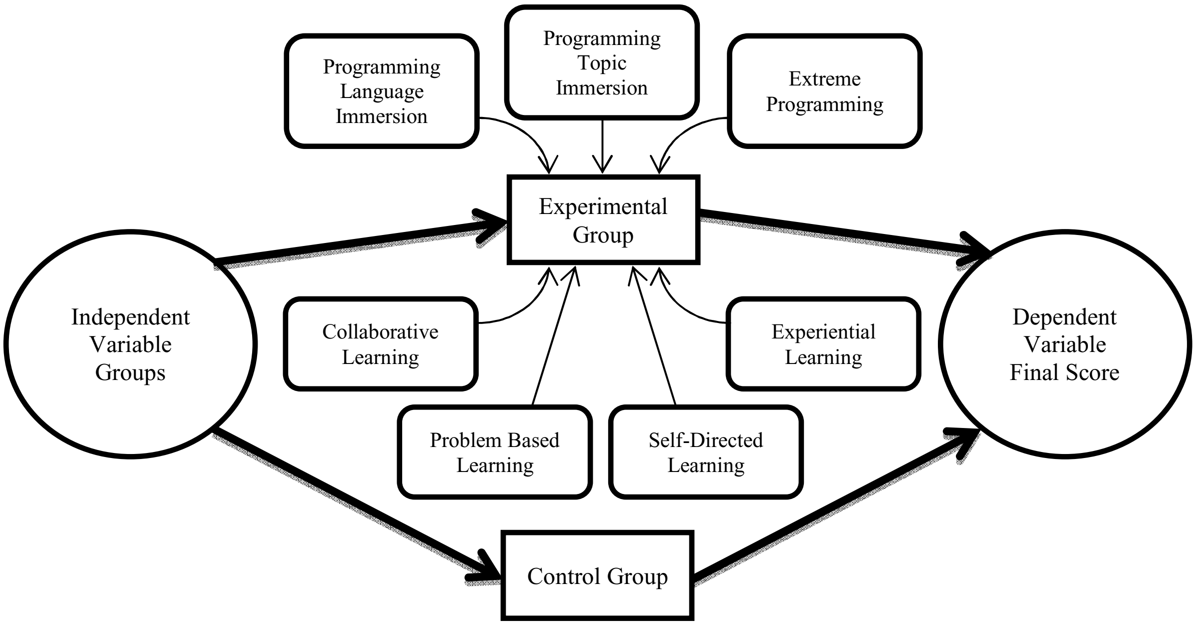

Figure 1. Theoretical Framework of the Study

Nature of the Study (or Theoretical/Conceptual Framework)

The study set out to verify whether a different approach for teaching computer programming to freshmen students can be as efficient as or better than the traditional approach used today in higher education institutions. The different and proposed approach was worked with the experimental group and the traditional approach was worked with the control group, as presented in Figure 1. The proposed approach was based on the implementation of a new strategy that combines ideas from different areas that are considered appropriate in supporting important educational theories and concepts. One of the most important ideas associated with the new strategy is the immersion language theory. Harper (2006) had proposed this theory to be useful for teaching computer programming languages. This research also used collaborative learning strategies in order to provide for an adequate experience of immersion.

The collaborative learning occurred initially at the whole group level. After that smaller teams (4-5) collaborated, but the goal was to be working as soon as possible in pairs. Eventually they worked alone for most of the circumstances. Any of the described collaborative setups can be used at any time depending on the circumstances. Pairs must be changing periodically in order to promote diversity that may be good for enrichment of each student's experience.

One element of the *immersion* experience was based on analyzing the statements and logic of an already working program. In implementing the approach the program will be initially run to understand what it does. After having a good understanding of the execution, the program will be analyzed statement by statement to understand what each statement does. The essential part of this activity is that the explanation of the purpose of each statement must emerge from the students, not from the professor. Students will be immersed in understanding the programming language statements on their own. This will be referred to as *programming language immersion*. The described process will be supported with the corresponding flowchart and pseudocode listing of the program, and Microsoft Excel will be used as a tool for running programs manually and for simulation of storage areas as occurred within the computer's memory.

Another element of *immersion* is directly related with the change of the general strategy used for defining the optimal order of the topics included in the course. Students will not be only presented initially with fully functional programs for analysis but with solutions to problems using building blocks and structures that are more complex than those used when working with the traditional approach. This is referred as *programming topic immersion*. In this, students are directed to analyze the topics under study using their own experience with computers. They have the opportunity to analyze real life situations that can help them in identifying essential structures like the selection control structure or the repetition control structure and how they are implemented in real programs.

After being exposed to the experiences described, the students work in the development of real programs, solving the problems presented, where they can use and implement the ideas they produced in discussion. This occur collaboratively in the beginning, with different setups, and alone later in the course. *Extreme programming* is used in this context leading students to work in two-person programming teams, working in implementing the solution in a short time, and testing at all stages of the development process. These are all characteristics of the *extreme programming* technique.

Due to the fact that students base their analysis of programs on previous experiences, and that they work immediately in producing real programs, they are immersed in *experiential learning*. As McKeachie and Svinicki (2006) established about *experiential learning*, "[w]hen students sense they are finding out something that is important and useful for themselves, using their own powers of observations and interpersonal skill, everything changes" (p. 278). It was expected that through the use of the presented methodology, students can be more engaged in the learning process making the process of acquiring needed programming skills easier for them. McKeachie and Svinicki addressed the importance of learning as a challenging process. They mentioned that, "[c]hallenge is a big source of motivation for students" (p. 137). These aspects are essential characteristics of the proposed approach. The *problem based learning* approach broadly used in education is also an important characteristic of the approach that is totally based on the solution of problems, an innate characteristic of the programming process, as described in Pokorny (2009). Finally, because students must analyze program code in order to understand the algorithms presented, they are developing important skills that promote in them the ability of being *self-directed learners*. This is an important characteristic associated with the assumptions of andragogy as specified by Merriam, Caffarella, and Baumgartner (2007) and Knowles, Holton, and Swanson (2011).

Organization of the Remainder of the Study

Chapter 2 provides an in depth overview of all the different aspects associated with the proposed approach for the study are discussed based on a review of literature. They are discussed in terms of the situations, studies and ideas that had served as promoters of the approach. Also, the situation of the research in the area of teaching computer programming is addressed mostly in terms of the different approaches used in a variety of circumstances, all of them trying to address effectively the main problem discussed for the study. However, as can be seen, a really effective solution had not been identified yet.

Chapter 3 is about the methodology to be used for the described study. The research design will be presented with details about the sample, the setting for the study, the instruments to be used, the data collection methods and analysis for both qualitative and quantitative areas, the treatment to be applied with the experimental group, and aspects associated with validity, reliability and ethical considerations associated with the study.

CHAPTER 2. LITERATURE REVIEW

Introduction

Teaching computer programming logic to computer sciences' students or to students on other Information Technology (IT) related areas is considered a real challenge by faculty members working in different educational institutions at different levels. Being successful in the process of providing the necessary programming logic skills to all students at the same time is virtually impossible. Researchers and faculty working in the referenced areas are constantly looking for ideas to become more effective in teaching these skills. As can be seen in the literature, some important advances had been made through the development of tools, techniques, methodologies or strategies that can be helpful in the teaching learning process of programming. Some important approaches were addressed and continue to be explored by researchers and they probably will continue to be developed in the future. They include the use of algorithm simulation software based on character representation, programming code simulation with immediate feedback, and the application of pedagogical methodologies in teaching programming.

Most of the ideas addressed, trying to work with this challenging situation, have important elements that can be combined effectively with others in order to identify new ways of teaching this type of knowledge and skills. The language immersion theory and the techniques used in extreme programming are some of those ideas that can be combined effectively with different educational theories and concepts in order to provide for a richer experience that can improve this teaching learning process. However, how challenging can be the teaching of programming, according to the experience and situations presented by the different authors?

Research into Teaching Programming

Teaching Programming or Programming Languages: A Real Challenge

For IT educators one of the biggest issues is finding better ways to teach programming logic to first year students. One technique that has been used for years is to introduce some programming logic ideas with the use of algorithm simulation software of different types. Some of them are based on the idea of representing characters that react to specific instructions provided by the user in the form of individual statements or as a group of statements. Another approach used is software that allows writing programs in a specific language, in pseudo code or by using other building blocks to simulate the execution of a program that allows creating, running and debugging them. Both of these approaches, and possibly others, have the same purpose, trying to develop programming logic in the student's mind in the easiest and more efficient way.

Teaching programming logic is a real challenge. This is not a simple task. It is associated with a variety of factors that can have a positive or negative impact in the process, depending on its characteristics and the students' specific situation with them. Some of the factors that can be important in this situation include the misconceptions of students about programming concepts or about the area of study they are enrolled in, the problems associated with the lack of good mathematical skills, and the natural complications of learning programming logic that can cause frustration to some students. Wiedenbeck, LaBelle, and Kain (2004) referred to different factors that were addressed by some researchers as important in determining the student's success in first year programming courses. These factors include student's previous experience with computers and programming, his or her mathematical and sciences background, student's learning styles, student's feelings of success to oneself, his or her feelings associated with outside forces, the course expectations, and self-efficacy. Cantwell-Wilson and Shrock (2001) mentioned as factors predictive for success in first year computer programming courses the following: "previous programming experience, previous non-programming experience, attribution for success/failure, self-efficacy, comfort level, encouragement from others, work style preference, math background, and gender" (p. 184).

In teaching computer programming to undergraduate students some factors cannot be controlled, such as those associated with the mathematical, sciences and computer background, the experience working with computers, or the wrong expectations made about the program due to an inappropriate orientation process. However, other factors may be influenced positively like the student's feelings of success, the self-efficacy and predisposition of the student to continue in a program, the student's self-motivation toward programming, the student comfort level, the student work style preference, and possibly others. Through the use of adequate teaching strategies and establishing an appropriate approach a teacher can develop a positive learning experience that can successfully support the whole process. The proposed approach is expected to be helpful in this process.

The tools and techniques mentioned to deal with the consequences of the referenced factors are part of the mechanisms explored by teachers and researchers. Others opted to work with well-known pedagogical techniques. A third approach analyzed involves combining pedagogical techniques with the use of the tools and techniques previously described. This third approach sounds to be more appropriate because it combine the best of both worlds. The proposed approach of the study is precisely a combination of different pedagogical strategies and theories with the tools and techniques associated with the language immersion theory and extreme programming (XP) techniques. Language immersion and the techniques from XP may use, support and are supported with the referred important pedagogical concepts, theories and ideas.

Following, the major approaches used in different research studies are presented. Which is the best one or the worst, this is hard to establish. Identifying the "best ones" or the "worst ones" may be in some way subjective, but it can be analyzed based on the efficiency or effectiveness of a specific tool. At the same time, the efficiency or effectiveness may be analyzed in terms of how well the tool helps learners to acquire programming logic techniques or how easy it was for the teachers to illustrate programming logic concepts in their classes. The chain continues with an effect in students' self-esteem and their feelings in terms of self-efficacy, having a final effect inclusively in their decision to continue or to leave the computing program.

Research Based on Programming Simulation with the Use of Characters

The simulation software based on characters has a basic organization composed of one or more characters that can be manipulated to execute different tasks in an environment that describes a specified scenario. The characters may be directed to execute specific tasks and, in some cases, interact with other characters or elements of the environment. The tasks are specified through statements, organized in an algorithm, forming the basis for solutions to specific problems.

Maureen Sprankle, a Professor Emeritus at the College of the Redwoods, in Eureka, CA, was the creator of Otto the Robot (Sprankle, 2005). Otto is an imaginary robot that has a set of commands with only 15 instructions that can be used to direct it to execute simple tasks, mostly related with the movement of its body parts. Additionally it has a memory space that can support only a single integer which may be incremented or decremented by one as desired. Some examples of the Otto's instructions statements are the following: *Stand up* (directing it to stand up from a chair in which it was previously seated), *Raise arms* (directing it to raise its arm in order to have the capacity to detect obstacles) and *Take a Step* (directing it to move forward one step). Basic memory commands included *Add one* (directing it to increment the number in memory by one) and *Is the number in your memory zero?* (directing it to check if the value stored in its memory is currently zero). Other statements allow the repeated execution of other commands and the verification for obstacles such as walls. It can be seen that Otto is a relatively simple tool.

What is the purpose of Otto? Otto was created to solve problems. These problems are made according to the capacity of Otto but someone must direct it adequately in order to succeed in completing the tasks related to the problem. That is the work of the students. They must write a simple algorithm or "program" that can direct Otto in solving the problem successfully.

According to Sprankle's (2005) methodology to teach programming logic concepts, Otto is the first encounter students have with structured programming. The only things they knew previously were general ideas in problem solving theory including the development of simple algorithms describing the solutions of simple problems from everyday life. This methodology seemed to be appropriate for many years in classrooms as she established. The faculty members from the Computer Science Department of the university in which the study was developed began using Otto before the year 2000 and used it until 2008 for teaching programming logic to first-year students with very good results. However, they began to analyze issues related to the effectiveness in continuing using it, or began looking for other approaches or tools that could be better, more advanced, more effective or simply more interesting for students to work with.

Some of this type of software received influences from previous tools like LOGO, a programming language created in 1967 that had a special tool named the Turtle. It allowed the creation of simple drawings on a two dimensional board area using simple instructions like those existing for Otto (Conway, Audia, Burnette, Cosgrove, & Christiansen, 2000).

As mentioned before, Otto is one of those tools that has the characteristics of algorithm simulation software based on characters representation. When Otto was used, this was done using a pencil and paper approach. As described before, it resulted in some way effective in helping give students an idea of programming logic concepts. Eventually some experimental versions of Otto were created as software tools. However, by that time there were other initiatives already running that resulted to be more attractive in general than Otto. One of the most important was the development of Karel the Robot, which could be seen as a near relative of Otto but with more options (Patfis, 1981). It sounds that Otto is younger than Karel but they are really contemporary.

Karel the robot was a software simulation tool since its creation. Becker (2001) made the following basic description of Karel's world:

> Karel inhabits a very simple world. There are avenues running north and south numbered one to infinity and streets running east and west, also numbered one to infinity. Walls may block avenues or streets. Beepers may be placed on the intersections of the avenues and streets. Several robots may exist within the same world.

Within this world, robots may move forward from one intersection to an adjacent intersection unless the way is blocked by a wall. They may turn left 90 degrees to face a different direction. Robots may pick up a beeper from their current intersection (if one or more are present), or place a beeper on their current intersection (provided they are carrying at least one beeper). Robots may carry beepers between intersections in their "beeper bag" and may detect whether or not their beeper bag is empty. (pp. 50-51)

Karel rapidly became a preferred tool for experimentation. It had a native Pascal like language that allowed, with simple commands, to control elements and functionality in Karel's world. Referring to this initial version of Karel and its use in classrooms by her students, Larason (1995) wrote that they "catch on quickly and start programming using a language similar to Pascal to maneuver Karel's movements" (p. 6). Additionally she argued that the use of Karel was fun providing too an interesting introduction to computer science concepts.

Becker (2001) referenced other versions of Karel that were developed with improved capabilities. For example, Karel++ based its functionality in the C++ programming language with object-oriented capabilities, but with important limitations in terms of the manipulation of variables and structures. Another Karel++ version was developed for Java by Joseph Bergin. Duane Buck, from Otterbein College, developed and integrated development environment that supported the programming of Karel the Robot with one of three languages: Java, Pascal, and Lisp. All of these versions, and others developed, represented simulation tools with limited language capabilities but with good possibilities for teaching introductory programming concepts. There was even a version of Karel based on functional programming. This use of programming languages will be addressed in the following section from the point of view of programming code simulation environments.

Sanders and Dorn (2003) presented Jeroo, a descendant from Karel and Jessica (a project developed by Lai Kuan Tong in 1990). Jeroo was created and tested at Northwest Missouri State University. They study its efficiency through a questionnaire submitted to students after the transition from Jeroo to Java occurred. They found a strong tendency in student satisfaction when using Jeroo to help them in the transition to Java programming.

There is an important tool that has acquired great importance in more recent years called Alice (Cliburn, 2008; Conway et al., 2000; Johnsgard & McDonald, 2008). The Alice project was developed by Conway et al. and represented a 3D environment with scenarios in which characters and other elements could be placed and manipulated through programming instructions. It had been considered an extraordinary tool for teaching programming language concepts to first year students. Conway et al. wrote that, even though it was "originally designed for undergraduates, we have observed that many middle and high school students are capable of using Alice to build interactive 3D graphics programs" (p. 492).

Cliburn (2008) presented the results of a study made with first year students about the effectiveness of using Alice to help them learning Java programming. They made a study using a survey in which they asked the students questions about the experiences they had using Alice and how it helped them in learning Java. The results they obtained were that 59.5% of the students considered that the tool helped them in learning Java versus 40.5% that not (p. T3B-4). However, the institution considered them not positive enough because of the high number of students that considered that Alice did not help them in learning this language (p. T3B-5).

Johnsgard and McDonald (2008) were also involved in a study about Alice with students of software engineering and computer science. In this study they evaluated the effects of using Alice in a more formal way because a comparison was established between the performances of students that did not use Alice versus those who learned programming concepts using this tool. From those who learned programming logic with Alice, 70.3% of students passed their first programming course with at least a C. From the group without learning supported by Alice, the number of students that passed the course was 46.4%. Additionally, the survey showed that, students "virtually all felt that the project in Alice was a valuable learning experience" (p. 134). Based on these results, the institution expressed a desire to continue using Alice as a tool for teaching programming concepts.

Greenfoot is a programming learning tool developed by Henriksen and Kölling (2004). It was based on all the existing tool characteristics of its predecessors in order to make it a better and easier to use product. It included ideas form Karel, Jeroo, BlueJ, Turtle Graphics, and Alice. One important characteristic it had was the availability of different scenarios. There was not any study found that evaluated the impact of the use of this tool in learning programming concepts.

A different situation was found with the work of de Kereki (2008). Her work included a formal study evaluating the benefits of using a tool known as Scratch. "Scratch is a new programming language that makes it easy to create your own interactive stories, animations, games, music, and art" (Scratch, 2008 as cited in de Kereki, 2008, p. T3B-8). Similarly to the other tools described before, Scratch allows to create one or more characters that can be manipulated through programming. Programming is made through the visual arrangement of blocks that allow the generation of a script to manipulate the characters created. The tool was previously proposed by Malan and Leitner (2007 as referenced by de Kereki, 2008) for teaching computer science programming courses but in a different setup to the one proposed by de Kereki.

Scratch was used by de Kereki (2008) in the first two weeks of the Computer Science 1 (CS1) course with freshmen students at the ORT Uruguay University. It was used to introduce initial concepts in the first week and more formal and complex structures and tasks in the second week. After the second week, the students moved to an actual programming language, in this case Java for one group and Visual Basic for the other. The study presented by de Kereki was a comparison between control groups and experimental groups. Control groups did not use Scratch and the experimental groups used Scratch. The two more important aspects evaluated were the perception of students of the educational experience in general and the retention rate of students after ending the first year of classes. Students in the experimental groups expressed having a significantly more positive experience in comparison to the control groups with over a 50% of difference between all students' means. On the other hand, there was not any real difference detected on the retention rate. The values were similar for all groups.

For de Kereki (2008), these results meant that the tool was good in motivating a good educational experience between students but this did not imply any effect in students' motivation to continue in the program. The author expressed the need to perform more research exploring different alternatives that could be more useful in evaluating the effectiveness of using Scratch.

Esteves et al. (2011) worked an important study using the action research methodology in which they analyzed the Second Life (SL) virtual world that is an environment similar to Alice. According to them SL is better than Alice because it "allows several users to connect, interact and collaborate simultaneously at the same time and in the same (virtual) space" (p. 626). This permits a collaborative use of SL. The importance of their study will be addressed in the following sections.

Overmars (2004), from Utrecht University, suggested using a game development tool such as Game Maker, to teach programming concepts to computer science students. Evaluating the educators experience with the integration of Game Maker, he said that "[g]enerally, students respond enthusiastically to these courses because they prefer programming in Game Maker to doing their usual assignments" (p. 83). Thomas et al. (2011) worked a study that precisely analyzed the use of game development for teaching of programming. The research was in the context of high school students learning computer programming and making games intended to be used by elementary school students. They found the idea useful in some aspects but negative in others. One important negative aspect was that not all students really participate in the creation process. They basically spent their time using the games developed by others. However, the methodology used had an important aspect associated with the extreme programming techniques and the agile software development methodology that will be discussed in the extreme programming section.

A common characteristic of all the simulation tools described, and the strategies used in implementing them, is that they expose learners to programming concepts early in the courses. This results sometimes in an improvement in motivation for students as established by the study from Sanders and Dorn (2003). Others studies demonstrated an improvement in the learning experience in general but with results that varies from softer to stronger in their level of significance. This can be viewed when comparing the results obtained from the study about the use of Alice worked by Cliburn (2008) versus those obtained from Johnsgard and McDonald

(2008). In this sense an expression were made about the need to work with more research in the evaluation of these tools and the strategies used with them.

The idea of having an early exposure to programming concepts was of importance in the use of these tools. It was the original intention in the use of Otto and Karel, and it continued to be use with the other tools created later, including the Alice, Scratch, and SL. This is an important idea in consideration as part of this dissertation research but from a different point of view. Instead of promoting an early exposure to programming logic through any of these tools, why not considering promoting that early exposure with a real programming language with real problems? This is an idea that was present in some way in the works presented by Overmars (2004), and Thomas et al. (2011) in which the development of games is the main purpose of the approaches used.

In this section an exploration was done of the tools that had been used for years by teachers to give their students a more effective experience in learning programming logic. Most of them resulted useful in different situations. It can be established that the efforts are valid and important advances were made but the optimal balance has not been found yet. In the following section another approach is considered. It is based in the use of tools that can accept programming code in simulation environments while providing immediate feedback to students. It will be seen that some of the approaches presented in this section are integrated with the approaches of some tools in the following section. This shows the importance of mixing efforts in favor of obtaining the best results from both worlds.

Research Based on Code Simulation with Immediate Feedback

Some advances had been found when evaluating possibilities to improve the efficiency in teaching programming logic to IT students using an approach based on the idea of simulation.

Researchers explored tools that allow for simulations based on code entered by the student. The most important feature of this idea is the immediate feedback that is received from the simulation tool. As Fernandez-Aleman (2010) stated, this type of tool allows for the evaluation of the correctness of computer programming code. This allows students to see immediately the results of their work, including the errors reported, and gives them the opportunity to immediately make adjustments. This is an important part of the learning process associated with acquiring programming logic skills.

Becker (2001) presented an improved version of Karel based on Java which included variables and standard structures used in the language. This allowed the teaching of object-oriented programming before introducing procedural programming concepts. The important difference between this version of Karel and the previous versions presented was that in this version a complete Java program was developed to manipulate Karel. In this situation the simulation takes place by executing the program with immediate feedback. If the program does not execute correctly the result is viewed immediately and the adjustments can be immediately worked.

Becker (2001) did not have a formal study of the use of this version of Karel, but he presented some informal opinions of a first implementation in a real course. He argued that the opinions were mostly positive between students. However, Becker presented a detailed work describing a methodology to use the tool in a first year course in computer science, and a deep analysis had been presented about its advantages in allowing teaching complex concepts, such as object-oriented programming, early in a course. According to this author, "[t]he result has been a course that is fun for both students and instructors, and where students understand the fundamental concepts early, allowing them to approach advanced topics with confidence" (p. 54).

Bergin (2006) addressed the advantages and disadvantages of Karel J Robot, another Java-based environment to control Karel simulation using the Java language syntax and its object-oriented features, through a drag-and-drop editor named Karel Universe. Karel Universe may be used totally by dragging and dropping fragments of code without needing to write the instructions, minimizing the possibilities of syntax errors. The blocks of code organized by the student can be executed after ending with the desired arrangement in order to view the simulation. This approach can be considered another good example of immediate feedback. However, a problem can be identified in the situation of not providing students with the opportunity to develop the code from scratch, since the tool was a graphical user interface with predefined blocks of code that must be arrange adequately to build a program. Students cannot have a real opportunity to learn easily the nuts and bolts of each the statements.

Borge, Fjuk, and Groven (2004) were involved in a more formal experimentation exercise based on interviews. They exposed students to Karel J (the J is for Java) for teaching object-oriented programming concepts at the beginning of a first year course but in a collaborative setup and only with two days of exposure. The results were positive. They demonstrated the effectiveness of using Karel J this way. They emphasized the important role played by the collaborative approach used because of the interaction between students by solving the problems together and helping each other in understanding complex concepts. Probably the most important conclusion of the study was about the successful experience experimented when the collaborative approach was implemented while using Karel J. This supports the intention of using extensively the collaborative learning strategy as part of this dissertation research.

However, as previously expressed, it may be unnecessary to spend some time in working with simulated environments like this in favor of having the possibility to analyze and develop programs with real life situations from the beginning.

In a different approach to programming code simulation tools, Cilliers, Calitz, and Greyling (2005), from Nelson Mandela Metropolitan University in South Africa, created an integrated development environment (IDE) named B#. B# had a special iconic programming notation that allowed students to create programs visually with flowchart-like symbols. They combined this IDE with Delphi—a Pascal-based language—to teach programming to first year computer science students. In a formal empirical study they evaluated the performance of students by classifying them in two groups: one using only Delphi and the other using Delphi and B#. From an original sample of 148 students with 60 participants in the experimental group and 88 in the control using stratified sample analysis, they finally worked with 59 participants for each group to maintain homogeneous sizes. Through statistical evaluation, the effectiveness of using the iconic technological support system in teaching programming over the traditional paper-based and language teaching approach was demonstrated. This conclusion was supported after using two tailed t-test and χ^2-test in the statistical analysis of six variables.

From the Cilliers, Calitz, and Greyling (2005) study it is important to analyze the function of having as part of the educational process in teaching computer programming courses of using what they named as visual iconic programming notation. They used B#, a tool of their own creation to work with the flowchart based representation of the programs in support of the learning process with argued good results. In this dissertation research it is considered an important element of the process the inclusion of the visual representation of programs through flowcharts as part of the proposed approach. The flowcharts will have an important role in the initial exposure of students to programming code.

Cheung, Ngai, Chan, and Lau (2009) presented a text-enhanced graphical programming environment named BrickLayer that allowed the construction of programs through the graphical arrangement of building blocks. This approach combined visual and textual elements and they evaluated if it was a better approach for teaching programming concepts in comparison to use only a programming language (in this case C). The great advantage of this tool was that it provided immediate feedback to students about the correctness of the code generated while visually constructing the program. They performed a study during four summer workshops with junior high-school students in which they taught programming concepts using BrickLayer to one group. Another group received the same instruction only with C. Based on the results obtained in the study, they argued that BrickLayer was an effective tool to teach programming. One of the things they measured was the impact of the workshop in influencing students' motivation to select a computing degree for their university studies. It was found that 83.3% of the students expressed their inclination to enroll in this type of degree after using the BrickLayer environment.

As discussed about the Karel J Robot presented by Bergin (2006), the use of BrickLayer has the problem of not allowing students to develop programming code from scratch. This situation may create inadequate mental mapping in students' minds about what they can really expect when they begin a degree in computer science or any other computer programming related area. This is in contrast to the situation of promoting an increase in motivation to continue studies in the area because of the effect produced by the tool in facilitating the programs development and having an immediate feedback.

Another tool with immediate feedback about the correctness of the developed programming code was developed by Truong, Roe, and Bancroft (2005). They proposed the use of an automatic program analysis tool that was worked through a framework with a "fill in the gap" capacity as part of a web system for teaching programming to first year IT students. They described its use for four years with 400 first-year students of the Queensland University of Technology in Australia. There was not any formal study developed about the effectiveness of the tool. However, they argued about its importance in providing immediate feedback while testing programming code with success according to their experience. They recognized that the tool had some limitations in specific circumstances but there is not any doubt about the importance of the idea of providing immediate feedback in any educational context. This is also crucial in providing an optimal educational experience in the teaching of programming.

Fernandez-Aleman (2010) presented a tool that was basically focused in the automatic assessment of programming code. Mooshak was a web based tool originally created to support computer programming contests. It was a free third generation tool already in use as an assessment system. It was adopted previously in supporting computer science courses. In these courses, Mooshak was adopted as an “automatic judging system for grading lab assignments and for self-assessment purposes” (p. 2).

The researcher argued that a positive difference existed in teaching computer programming with the support of Mooshak and other web related tools versus teaching it using the traditional approach. In three of the four aspects he evaluated, they found that the difference was significant, thus supporting the hypothesis presented in favor of the use of Mooshak as a tool in teaching computer programming concepts. The three aspects that resulted in a significant difference were in the assessment of problems associated with debugging (identifying diverse types of errors existing in a program), deployment (preparing automatic installation files for new software) and versioning (managing the files of software necessary for identifying differences between versions). The difference in testing (making runs of software to validate its functionality using different techniques) resulted not to be significant.

The importance of the ideas presented by Fernandez-Aleman (2010) are in some part related to the importance of the immediate feedback as established previously in this section in analyzing the works of other researchers. Also another aspect of importance is in the reference he did about the use of techniques traditionally employed in the development of software, as supporting processes for teaching programming. Testing and debugging are processes that will be definitely part of the proposed approach in supporting the development process and in providing immediate feedback to students when working with the programming exercises to be developed.

In this section important advances in the area of code simulation tools that provide immediate feedback were discussed. As mentioned, the existence of such tools is considered very important in the learning process associated with computer programming skills acquisition. This is generally associated with the use of appropriate methodologies and techniques in terms of the pedagogical process involved in teaching these courses. The following section considers pedagogical approaches evaluated by researchers in working computer programming courses.

Research about the Appropriate use of Pedagogical Techniques

The integration of pedagogical concepts in computer programming logic courses has been considered by many professors in a variety of circumstances. However, there is a lack of studies in this area. This was addressed by Sheard, Simon, Hamilton, and Lönnberg (2009) (2009). Sheard et al. presented a study they conducted analyzing the research literature available in the teaching and learning of programming. An increased interest was detected in the research associated with the teaching and learning of computer programming issues. However, as mentioned, there was a lack of studies specifically centered in the pedagogical aspects associated with programming. This establishes that it is extremely important to promote the development of more research in the area.

The Sheard et al. (2009) work was very useful and important for IT and the associated educational context. It provided a good idea of the status of research in the area. However, there is significantly more work that needs to be done. Suggesting this need for more attention in research in the area, Merriam, Caffarella and Baumgartner (2007) discussed about the applicability to programming courses of the theories about adult education. They referred, for example, to the theories and concepts associated with self-directed learning.

The Staged Self-Directed Learning (SSDL) instructional model presented by Grow (1991) and Grow (1994) was directly associated with self-directed learning. SSDL included what Grow named the four stages of learners that may be used in teaching, helping students to have a richer learning experience. The stages identified were the dependent learner, the interested learner, the involved learner, and the self-directed learner. With this model, teachers can provide students with the opportunity to go faster in a class without waiting for other students in the class. Using this model in programming classes may be very important and useful. Students can learn programming knowledge and skills in different ways and at different times. This possibility may be very important for different courses in computing areas.

In a unique article found about the application of pedagogical concepts in the teaching of computer programming, Ali (2005) referred to this in the context of students at the undergraduate level. He presented arguments about the most effective way to teach programming concepts. Critical thinking was identified as an important part of this process. Some strategies were referred as part of this like scaffolding, concept mapping, and constructivism. Also other strategies to be use included problem based learning, collaborative learning, and active learning. In Table 1 all of these strategies can be seen with their corresponding descriptions. Ali did not reference any formal study about the use of these strategies. However, he argued that in his own experience in using those strategies in teaching computer programming classes to computer science's students their effectiveness were demonstrated.

Ali (2005) showed a serious reflection about the use of known pedagogical strategies in the teaching of computer programming. Doing a really good job, Ali described important aspects of the teaching process that are frequently used in different situations but are not normally identified with a specific name. It can be viewed that some of the strategies referenced are present in different circumstances of all the studies and works discussed. In the specific context of the proposed approach for this dissertation research five of the strategies described are considered important parts of it. Scaffolding, constructivism, problem based learning, collaborative learning, and active learning are all part the strategies to be used in implementing the proposed approach, and all of them are essentially important for the whole process. The success of the proposed approach relies in great part in the presence and effective use of these strategies. Table 1 shows a summary of the strategies presented by Ali.

As established, there is a lack of research addressing pedagogical issues in the context of teaching computer programming courses to students of IT and other related areas. There is a need for promoting this type of study because it can be helpful in providing more optimal and complete experiences to students. The final result can be beneficial to students and faculty, and incidentally to academic programs in the computing area. By using the appropriate strategies, the possibility of having students motivated with programming and, at the same time, predisposed to continue in computing programs is higher, probably allowing better retention rates in these programs. Studies in these areas will be also important in supporting the different ideas developed by teachers and researchers like those discussed previously. Evidence of this is the presence of the strategies discussed by Ali (2005) in some of those works discussed previously.

Table 1. Strategies for Teaching Computer Sciences Courses as Referred by Ali (2005)

Strategies	Descriptions
Scaffolding	It refers to a strategy of teacher support that "is gradually removed as the learners become self-reliant" (Ali, 1995, p. 246). It implies the need to locate learners' needs and to provide them support dynamically.
Concept mapping	The concept maps are helpful in representing knowledge conceptually through a two dimensional diagrammatic representation. Nodes are used to represent concepts, and links between nodes are used to represent relationships between concepts helping to show ideas, design complex structures, and assess understanding, among others.
Constructivism	A view of knowledge acquisition in which new knowledge is generated through the interaction of new ideas with things already existing in learner's mind. Final understanding of new concepts occurs when the new information is assimilated with the old previously acquired from different experiences.
Problem based learning	Through the use of problems designed to students in problem solving skills and critical thinking skills, teachers facilitates progressively and environment of self-learning giving to learners the responsibility for their education.
Collaborative learning	It is simply the idea of learning in groups with the observed advantages of learning more quickly, sharing ideas using adequate terminology, facilitation of teacher's role as observer, and having students more relax and friendly.
Active learning	It implies the students' involvement in classroom activities that are more active than simply been listening to lectures. Some activities are discussing, problem solving, reading, and writing.

An Important Discovery from the Use of Otto

In the Sprankle (2005) methodology to teach programming logic concepts, Otto was defined as the first encounter students have with structured programming logic routines. Their prior knowledge consisted of general ideas in problem solving theory including the development of simple algorithms describing the solutions of simple problems from everyday life in plain English. Otto immediately faced students with structures that are essential for the programming process in a very early stage of the courses. The most important structures included the sequential control structure, the selection control structure, and the repetition control structure. They are the most simple and essential ones in programming.

The success of Otto while using the context of the first weeks of programming courses, and the success achieved in using other tools based on similar characters, established a basic idea. It is that most students have the capacity to understand these structures in a very early stage of the courses. As students can develop simple routines to solve a problem giving directions to Otto using the three structures without having major problems, it is suspected that they can have the capacity to understand and to work with these structures in different contexts. This may be achieved using techniques associated with the language immersion theory, extreme programming, and important ideas from education.

Language Immersion and Extreme Programming

Language Immersion Theory

The language immersion theory had been addressed extensively in the literature from the point of view of its use in promoting more effective ways to teach nonnative languages. Immersion language programs are formally implementing this theory in diverse situations and for different languages as described in Cobb, Vega, and Kronauge (2006), Yamauchi, Lau-

Smith, and Luning (2008), Cheng, Li, Kirby, Qiang, and Wade-Woolley (2010), Wesely (2010), Qiang, H. and Kang, Y. (2011), Usborne, Peck, Smith, and Taylor (2011), and Platt (2012).

Immersion language, as described by Harper (2006), is considered a technique based on leading students to speak only the language to be learned from the first day of class. The application of this technique includes a careful planning of the strategy to be used in each class. These ideas have their origin in theories presented by Jean Piaget.

Cobb, Vega, and Kronauge (2006) analyzed a two way bilingual program using language immersion. The program was based on providing a mixed experience of immersion to elementary school students. In the specific context used for the study they found that the education process reflected a real improvement in the immersion experience in comparison to the experience provided to students learning with the traditional teaching approach. The new approach used resulted more beneficial to both English and Spanish native language students. They recognized that the results obtained cannot be easily generalized to other contexts.

A language immersion program within the context of an elementary school was studied by Wesely (2010) through an important mixed method study that included 131 students responding to surveys and 33 students interviewed. In the study the author focused in determining if there was an important difference in students' motivation toward learning different areas using other languages (in this case Spanish or French). The study was done after students ended the elementary school with the immerse experience. Wesely compared between students that opted to continue in an immersed program and those that not. A real difference was not identified thus the immersion experience was considered good. Differences existed only in the individual perception of students of the experience.

Platt (2012) discussed how institutions are establishing language immersion programs in order to attend adequately the needs of a globalized world. This is more important in the context of programs like the one specifically described, in which they prepare graduate students that must be internationally competent in their area of study associated with providing psychological clinical services. The immersion experience was emphasized with Spanish native speakers in order to attend Latino communities in Mexico and in United States. Platt established that from a cultural perspective the immersion process occurs basically by promoting the development of relationships and direct dialogues with the people that are the target of the immersion process, in this case Spanish speakers.

Yamauchi, Lau-Smith, and Luning (2008) presented a study associated with the use of language immersion in teaching English to Hawaiian students. The study discussed was more focus in the level of involvement of parents in the language education process of their children. Cheng et al. (2010) worked a very complete study analyzing the use of language immersion versus a traditional approach in teaching English and mathematics to Chinese native language elementary school students. With very large samples of students they found that students working with the immersion approach definitely learned the English language better. Qiang and Kang (2011) addressed how the immersion language approach is being spreading as the preferred methodology for teaching of English in China. An extensive study was presented addressing many important issues associated with its use and the implications of its successful implementation for the country. Usborne et al. (2011) presented another example of successful implementation of the use of language immersion but in this case to students in Canada with knowledge of English, learning an aboriginal language named Mi'kmaq. The experience was demonstrated to be better with language immersion in comparison to the traditional established approach.

Harper (2006) presented an interesting situation in which a personal experience led him to change the way he usually worked in teaching computer programming. He took a Spanish course in which language immersion was used. The experience of Harper in learning Spanish prompted him to use a similar approach in teaching computer programming. He opted to face students with programming code from the first class.

The methodology being used by Harper (2006) was basically the same as the natural language immersion approach. Programs were going to be developed immediately with the teacher's guidance. In a first exposure to code, Harper directed students to read lines of code and to try explaining what they meant while giving appropriate support in order to avoid the students feel embarrassed by the situation. It was argued that the methodology could be implemented with a careful process of planning. Harper experimented with the methodology with real students and taking into account a variety of situations. However, he did not perform a formal study in order to evaluate the effectiveness of the methodology. This was part of his plans for future studies.

The study presented by Esteves et al. (2011) using the action research methodology resulted to be particularly important in terms of the aspects associated with the language immersion theory. In their study the immersion idea was directly addressed describing that through working with the Second Life (SL) virtual environment students were involved in a real immerse situation while they are trying to develop the different objects and activities in collaboration with others, and learning the tool at the same time.

This idea is important because it established that essentially all, or at least the majority, of the experiences lived by students in using the described tools can be associated with some level of immersion. Based on this it can be argued that the idea of immersion can be considered effective in most situations while using the described tools that resulted to be effective. However, the language immersion experience proposed for this dissertation research is different when compared to those used in the referred tools. On the other hand, the success demonstrated in using the language immersion approach in the teaching of spoken languages can be evaluated in order to determine if the same success can be achieved in the context of teaching computer programming in a specific programming language. This may be possible if the language immersion techniques associated are defined and used adequately.

Extreme Programming

Extreme programming (XP) "is the most popular of the various types of 'agile' software methodologies" (Agarwal & Umphress, 2008, p. 82) that are in use by developers. XP methodology is supported by a group of practices that give to it the base that was essential in its success and continuous growth. These practices are thoroughly discussed in Kobayashi, Kawabata, Sakai, and Parkinson (2006). Developers in organizations are adopting this methodology because of its flexibility in adapting for changes throughout the development process as oppose to models like Waterfall in which planning and design must occur previously to developing and implementing any code. As Agarwal and Umphress (2008) stated citing Wells (2003), "[e]xtreme Programming (XP) was created in response to problem domains where requirements change frequently" (p. 82).

As mentioned, XP is a programming development approach considered part of the agile software development approach. Agile in this context "means to strip away as much of the heaviness, commonly associated with traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines, and the like" as defined by Erickson, Lyytinen, and Siau (2005, p. 89). They defined XP as coding according to customer's specifications, and testing the code ensuring that prior steps are aligned with the intended development. They also referred to the twelve core practices of XP as: "the planning game, small releases, metaphors, simple design, testing, refactoring, pair programming, collective ownership, continuous integration, 40-hour weeks, on-site customers, and coding standards" (Beck, 1999; Jeffries, 2001; Wake, 1999 as cited by Erickson, Lyytinen, and Siau, 2005, p. 90).

Erickson, Lyytinen, and Siau (2005) presented a literature review of the research about agile software development and extreme programming in the last years before 2005. They noted that there is not a good amount of research in the area and that most of the available was about the XP, pair programming practice. The importance of developing more research in the area is strongly argued by them.

Mangalaraj, Mahapatra, and Nerur (2009) referred to the lack of enough research in the area of agile methodologies. However, XP was recognized as the most ascendant of the approach associated with these methodologies and an analysis was done by them through a case study about its acceptance and use within organizations that recognized its benefits for the software development process.

With very positive results Abdullah, Idris, and Subramaniam (2006) analyzed through a research study the use of an innovative technique to teach programming. The technique was the use of pair programming but implemented in an online learning environment. That was the reason to name it as virtual pair programming. Over 100 students commented on the benefits of using virtual pair programming with important references to the advantage of having the opportunity to collaborate with others. Hadar, Sherman, and Hazzan (2009) also discussed the importance of implementing collaborative tools and activities to the teaching process, associated with XP. Four attributes were addressed as been important parts of the process that are action learning, cooperation, problem solving, and reflection. Goel and Kathuria (2010) referred to the need to integrate in the teaching process of computer programming the XP approach of collaboration through pair programming. They established that students need to produce programs that can be understood by others. A research study was worked with 75 students. The results from questionnaires and interviews demonstrated some of the advantages of the methodology. They were an increase in efficiency, an increase in trust and improved team work, the development of problem solving skills, and an increase in quality of programs.

In the methodology studied by Thomas et al. (2011) about using game development to teach computer programming to high school students, some techniques of extreme programming and the agile methodologies were included. One of them was the development of games rapidly using the tools available to this purpose. Another was the collaborative characteristic of the experience in which students work basically in groups. A third one was having final users (elementary school children) to be part of the experience giving feedback about the games constantly to developers. In extreme programming this allows developers to integrate adjustments to the programs developed, immediately making the whole process more rapid and sometimes more precise.

Education Theories and Associated Concepts

Clarity and Initial Motivation of Students

In any course teaching learning process, providing an adequate introduction to the course, and to the different topics and units, it is essentially important in giving students a clear view of what they will encounter throughout the course. The introductions are the better place to give the initial motivation for the specific unit trying to capture the students' interest in the topics to be treated.

Having clear introductions is an essential aspect to be included as part of any instructional plan. Well prepared introductions can be helpful not only to serve as motivational statements but as tools to establish strong links with previous knowledge for students. Through this "[c]larity is achieved when the instructor provides a way for the flow of his knowledge to firmly connect with the neural networks of the learner" (Wlodkowski, 2008, p. 79). If according to the general characteristics of the students, the instructor is capable of addressing aspects associated to previous knowledge they are supposed to have, the probability for better engagement of learners with the new concepts increase significantly. In the study, trying to relate ideas for the course with previous knowledge will be essentially important to succeed in working with the proposed approach.

Historical aspects in the topics associated with the courses are important to be addressed, giving opportunity to students to think about it and to express ideas that can emerge in the process. This technique was presented by McKeachie and Svinicki (2006, p. 46). This may be helpful in leading to discuss different aspects of the course and in increasing students' curiosity toward the new concepts and skills to be acquired. Strategy 33 of Wlodkowski (2008) established that we must try to "Relate Learning to Individual Interests, Concerns, and Values"

(p. 249). It will be of importance in the proposed teaching approach to establish in students the pertinence of programming development in people's everyday life, including their own.

Being better in teaching implies getting the best of the different theories or methodologies, considering the aspects associated with the way the brain works in learning, and keeping students motivated (Wlodkowski, 2008). Also it is important to assure having the adequate expertise, empathy, enthusiasm, clarity and cultural responsiveness (p. 50) in order to provide for the best conditions for future development of students, inclusively in the lifelong context.

Course Goals and Objectives

Wlodkowski (2008) established through Strategy 6 the need to "clearly identify the learning objectives and goals for instruction" (p. 152). The goals and objectives are the simplest way students have to visualize what they are supposed to know and what skills they need to acquire. They must be clearly established because this "can heighten learners' sense of control and capability" (p. 153). Goals and objectives must be adequately constructed following the Bloom's Taxonomy (Wlodkowski, 2008; McKeachie & Svinicki, 2006). Goals and objectives for the course of the study must be highly oriented to problem solving, due to the specific characteristics of an introductory computer programming course. Schön (1987) argued that the problem solving goals are significantly different from the conventional instructional objectives.

The goals and objectives focusing on problem solving is not a strange situation because the essence of computer programming is precisely the solution of algorithmic problems. These are problems that can be solved through the use of computers. It is then obvious to work the course solving problems. Problem Based Learning is a methodology referred in the literature as "one of the most important developments in contemporary higher education" (McKeachie &

Svinicki, 2006, p. 221). As McKeachie and Svinicki (2006) wrote it "is based on the assumptions that human beings evolved as individuals who are motivated to solve problems, and that problem solvers will seek and learn whatever knowledge is needed for successful problem solving" (p. 222). A computer programmer should be motivated to solve problems. A student in this area that does not feel this motivation cannot be a good programmer. Through this motivation and the process of solving the presented problem they will receive the needed knowledge and skills that will help them to solve future problems they encounter. They are in continuous training acquiring new experience.

In these courses the problems students will be working on will be simulations of real life situations. By using realistic situations we can make the learning experience more motivational for students. Knowles, Holton, and Swanson (2011) mentioned that the motivation to learn is one of the main principles of andragogy.

Teaching Strategies

McKeachie and Svinicki (2006) suggested that "to record and remember materials effectively, you could use Microsoft Power Point to improve readability of lecture topics" (p. 233). In courses generally this tool is essentially important. It can be used in lectures, discussions, demonstrations and guided practices.

As teachers there is a need to assume the role of facilitators. This means guiding students in the teaching learning process, but leaving to them the opportunity to develop their knowledge with a minimum of intervention, promoting in them self-directness. However, there are situations in which teachers need to be more in charge of the process. It is important to strike a balance in order to be successful in the education process.

In computing course the use of guided practice and, after that, leading students to work individual exercises is essentially important. Exercises in this context follow in general the idea of the anchor instruction method into the experiential learning approach. Students will work with real situations that can be solved with programming. Working on problems, students are involved in experiences that can help them grow toward being experts (Merriam, Caffarella & Baumgartner, 2007, p. 183). This is associated with Strategy 51 of Wlodkowski (2008) that suggested to "Use Authentic Performance Tasks to Deepen New Learning and Help Learners Proficiently Apply This Learning to Their Real Lives" (p. 326).

The combination of all of these experiences helps students in developing skills that will help them to be self-directed learners because through these types of exercises they have the opportunity to correct their work. They can apply self-assessment. In a more general context, they are learning by doing. They also have the opportunity to be engaged in self-reflection about what they know and how can they solve the problems presented.

The use of different types of evaluation techniques is important to attend differences in learning styles and diversity. Brookfield (2006) referred to the characteristics of helpful evaluations, and he addressed that the different types of evaluations must be clear, immediate, regular, individualized, educative and future oriented, among others.

Andragogy and the Self-Directed Learning Assumption

As a good practice in computer programming courses, students must be required to complete some tasks with the least intervention of the professor. The idea is to force them to be self-directed learners. This is part of an important assumption in the theory of andragogy presented by Knowles (Knowles, Holton, & Swanson, 2011). An advantage of promoting the self-directed learning is that for learners the experience represents one that promotes self-motivation and self-discipline (Palloff & Pratt, 2007). Strategy 16 from Wlodkowski (2008) established the need to "[p]romote learners' personal control of learning" (p. 189).

Problem Based Learning and the Use of Real World Problems

Problem based learning is a technique presented by McKeachie and Svinicki (2006). About this the work to do by the groups is related to the following statement "[i]f realistic, relevant problem is presented… students will identify needed information and be motivated to learn it" (p. 222). Another important benefit with this is that they will engage in collaborative learning through the process of discussing with partners about the strategies to be followed.

Introductory programming courses must be taught with high use of technology because three of the goals established clearly specify the requirement to "develop", "work with" and "implement". This implies the highly practical and problem based approach of these courses. "Students must make decisions, solve problems, and react to the results of their decisions" but using technology at all times (McKeachie & Svinicki, 2006, 226). McKeachie and Svinicki (2006) argued also about the "need to consider which applications are appropriate for your students, course content, and teaching style" (p. 238). In Thomas et al. (2011) the study discussed was about an introductory course in programming to high school students using game development for teaching as a real life experience in problem solving. In this case the problems were the development of appropriate games for elementary school children with constant feedback about the results obtained from the young final users.

CIQ Instrument for Formative Assessment and Self-Assessment

Self-motivation can lead to more interest in new ways of learning and lead to continuing development. These processes imply reflection. The Critical Incident Questionnaire (CIQ) instrument developed and referred to by Brookfield (1995) allows working with this type of reflection through a formative assessment process. Reflections will be done on the best and most useful concepts they learned weekly, and on the worst and less useful things they found in the classes. This can be useful for students because this “sets the stage for participants to become lifelong learners” (Palloff & Pratt, 2007, p. 186).

As Palloff and Pratt (2007) wrote, “[a]ssessment should not be seen as a separate, cumbersome task but should flow from course activities” (p. 216). That is precisely what occurs when students are instructed about a work they must do including the checklist or rubric that will be used for evaluation, or when they are required to use the CIQ. The CIQ can be part of the requirements for the class but at the same time is the perfect guide for learners to be involved in self-assessment. At the same time they will have the opportunity to be in control of their own progress that is an important thing to be considered by teachers (Brookfield, 1995).

Collaborative Learning

Working collaboratively in educational setups is recognized as an important strategy that can be helpful for students in promoting better learning experiences than using other approaches like purely based on conferences and individual work. Inclusively in modern educational setups like those worked online or blended (with online and onsite activities) it is established the importance to work collaboratively. This is extensively discussed by Palloff and Pratt (2007), and is applicable to all levels and activities in the educational process, from lessons to assessment. For example, about the assessment process in terms of assignments online they suggested that “[c]ollaborative assignments should be assessed collaboratively” (p. 214).

It is important also to provide also possibilities for self-evaluation in the collaborative experience. As Wlodkowski (2008) in Strategy 54 for teaching and assessment, in the group’s experience the suggestion is to “Use Self-Assessment Methods to Improve Learning and to

Provide Learners with the Opportunity to Construct Relevant Insights and Connections" (p. 344). This may be very important in terms how much the students are involved in situations promoting experiential learning and, inclusively, transformational learning.

In a simple study presented by Othman and Othman (2012), they referred to the preference of students to work collaboratively in an online computer programming course. They implemented collaborative learning in teaching programming using the think-pair-share technique that is similar to the concept of pair programming used in implementing extreme programming. Goel and Kathuria (2010) also implemented collaborative pair programming for the teaching of computer programming. They argued about obtained effective results in different aspects like the general efficiency of the approach, trust and teamwork helping in clarifying doubts while sharing, problem solving skills improvement, and quality of learning.

Using the Second Life (SL) virtual world, Esteves et al. (2011) also implemented a collaborative approach in supporting the teaching of computer programming. In the study worked they discovered that there exist pros and cons in using the SL and implementing the collaborative approach. From this they insisted in the importance of having well defined projects in order to implement the approach effectively.

As can be seen the use of collaborative learning may have general positive implications in the learning process for students. In the literature exists a diversity of situations that demonstrate this. The consequences are important in providing students with rich experiences that improve the learning process allowing inclusively to promote having real situations of transformative learning.

Experiential Learning

The experience of learning using the proposed approach will engage students in experiential learning and probably in some level of transformative learning as well. As established the collaborative learning strategy will be an important element to be used in the implementation of the proposed approach. In the process they will be literally “learning by doing”. This is the main postulate of experiential learning. All the collaborative setup for group activities is related to the Strategy 11 presented by Wlodkowski (2008) who wrote, “Acknowledge Different Ways of Knowing, Different Languages, and Different Levels of Knowledge or Skill among Learners” (p. 167). This established the importance of the approach also in working with the differences in learning styles existing between students.

Transformative Learning

Merriam et al. (2007) affirmed that transformational learning implies a “dramatic, fundamental change in the way we see ourselves and the world in which we live” (p. 130). In some ways the theory of transformative learning describes “how learners construe, validate and reformulate the meaning of their experiences (Imel, 1998). Both of these statements have a strong implication with the idea of how this way of learning requires in-depth personal exploration. This exploration occurs through a process that includes four main components: experience, critical reflection, reflective discourse, and action (Merriam et al., 2007, p. 134). As Palloff and Pratt (2007) established, [t]his type of learning is rooted in the meaning-making process that is central to constructivism” (p. 185).

In a typical classroom setting the teacher can foster transformative learning by “establishing an environment that builds trust and care and” facilitating “the development of sensitive relationships among learners” (Imel, 1998). By providing a trusting environment teachers can promote autonomy and collaboration between students. The teacher must develop activities that can lead students to explore their own perspectives looking for new ways of thinking and motivating them to critically reflect on their experiences. It is important to motivate them to associate their critical reflection with affective learning (Taylor, 2000). From the reflection process students must be directed to talk with others about the new perspectives they developed.

The collaborative learning experience will engage students in the needed situations to develop precisely these reflective processes while collaborating with others. They will be involved also, in trust relationships with partners and with the teacher that is an important element of the transformative experience, as described. They will work constantly reflecting on the situations that occur, leading them to talk about ways to solve them. As established before, part of the idea of collaborative activities is to promote in them to be self-directed learners with only the support of partners. All of these are important aspects of the proposed approach.

The Proposal of a New Approach for Teaching Programming

A new approach to teach computer programming is proposed based on the integration of the described techniques of immersion language, extreme programming, and some theories and concepts from education. The importance of these areas were addressed emphasizing the aspects that will be directly associated with the proposed approach. Specific ideas of importance for the implementation include:

- The application of the immersion language methodology originally developed for the teaching of a second language, for the teaching computer programming languages (in this case Java).
- The implementation of immersion language in two different contexts:

- Programming language immersion – Directing students to analyze and to discover themselves the meaning and functionality of Java statements within the context of a running program. Based on these the will begin experimenting with the programming code in order to continue discovering their use and how they can be combined to build programs.
- Programming topic immersion – Students will be confronted from the beginning with building blocks and structures that are normally used later in classes using the traditional approach. They must analyze the programs based on previous experiences and knowledge from their lives and the work done in previous classes.

- Extreme programming techniques will be used in support of the immersion process. Students will be leaded to work in two person programming teams, working in implementing the solution in a short time, and testing programs at all stages of the development process.
- The collaboration at different levels (collaborative learning) will be an essential element of the class in which the pair programming technique will be the most used. However, other setups will be explored as needed according to the circumstances. The scaffolding strategy will be constantly part of the learning process. After having important experiences of collaboration, it is expected that each student can reach the goal of working alone in the development of programs.
- The course will be based on the implementation of the problem based learning approach as is common in this type of programming courses.

- The active learning process developed by the implementation of the proposed approach is expected to help students in being self-directed learners, in having richer learning experiences (experiential learning), and inclusively to be engaged in transformational learning.
- The CIQ will be used in support of the learning process helping students to be involved in self-assessment and allowing the professor to work with formative assessment. This can be beneficial because the professor will have the opportunity to make adjustments in the class in order to attend misunderstandings detected in previous meetings after analyzing the results of the instruments.

Most of these aspects were identified in the literature reviewed. Their importance and effect in different situations was analyzed and compared to the ideas associated with their used in the implementation of the proposed approach. In Table 2 the most important aspects associated with the implementation of the new approach are identified as they appeared as part of the different articles reviewed. The three main areas included are those associated with immersion and extreme programming (XP), those associated with the identified educational strategies and concepts, and those that addressed some research aspects in support of the methodology proposed for the study.

From the implementation of the proposed approach it is expected that the main situation of having problems in teaching computer programming logic to freshmen students can be substantially improved. This improvement is expected to occur in the context of a course offered making immediate use of the programming language to be taught, in this case Java, and using examples directly associated with the solution of real problems. This means that it will not be necessary to use any tool like those described in the literature as mechanisms to give students a first experience in the development of algorithms to solve unreal problems. The use of such tools implies that there will be a need to move from their use, and after being comfortable, to begin working with the real language of use. The effect is having less contact hours with the use of the real language because some weeks were spent playing with the referred tools. This situation will not exist in the implementation of the proposed approach.

Table 2. Aspects Associated with the Proposed Approach Identified in the Articles Reviewed

	Immersion and XP		Educational strategies or concepts						Research methodology	
Article	Some level of immersion	Can be associated to XP	Collaborative learning	Self-directed learning	Problem based learning	Scaffolding	Constructivism	Immediate feedback	Mixed method research	Experimental design
Sprankle (2005)	x				x					
Becker (2001)	x				x			x		
Sanders and Dorn (2003)	x				x					x
Cliburn (2008)	x				x			x		
Johnsgard and McDonald (2008)	x				x			x		x
Henriksen and Kölling (2004)	x				x					
de Kereki (2008)	x				x			x	x	x
Esteves et al. (2011)	x				x			x	x	
Overmars (2004)		x			x		x			
Thomas et al. (2011)		x	x	x	x		x			

Table 3. Aspects Associated with the Proposed Approach Identified in the Articles Reviewed (Continued)

Article	Immersion and XP		Educational strategies or concepts						Research methodology	
	Some level of immersion	Can be associated to XP	Collaborative learning	Self-directed learning	Problem based learning	Scaffolding	Constructivism	Immediate feedback	Mixed method research	Experimental design
Fernandez-Aleman (2010)	x				x			x		
Bergin (2006)	x				x			x		
Borge, Fjuk, and Groven (2004)	x	x	x		x			x	x	
Cilliers, Calitz, and Greyling (2005)				x	x	x	x	x	x	x
Cheung, Ngai, Chan, and Lau (2009)	x				x		x	x		x
Truong, Roe, and Bancroft (2005)					x			x		
Grow (1991) and Grow (1994)				x	x	x	x			
Ali (2005)			x	x	x	x	x			
Harper (2006)	x*	x			x	x	x			
Abdullah, Idris, and Subramaniam (2006)		x	x		x					
Goel and Kathuria (2010)		x	x		x				x	

* Proposed the idea of language immersion for teaching of computer programming.

CHAPTER 3. METHODOLOGY

Purpose of the Research

The purpose of this study is to examine if the implementation of a new approach for teaching computer programming to freshmen students in computer sciences causes an improvement in the acquisition of programming knowledge and skills, in comparison to students that receive the classes using the established traditional approach. It was expected that this new approach can result in a faster acquisition of the needed skills and in producing more confidence in students in programming code development. The new teaching approach was expected to demonstrate an improvement in students' performance at the end of the initial course. The improvement in efficiency obtained in solving computing problems through programming was expected to result in motivation for students to continue in the computer science program.

The established methodology, within the context of the study, was based in an incremental approach beginning with the simplest ideas and ending with the more complex ones. The proposed approach followed a different strategy using some of the existing ideas but supporting them with techniques obtained from the immersion language theory, extreme programming, and some educational theories.

From the immersion language theory the main idea was associated with developing an experience of total immersion with the computing programming language, even from the initial topics of the course, resolving problems that were considered adequate for the experience. Harper (2006), for example, proposed leading students to read and interpret lines of codes from an existing program sharing their ideas with the whole group instead of assuming the teacher's traditional role of reading and explaining the code to students in lecture. Similarly, students can be exposed to real programming exercises implementing the ideas explored in the discussions generated trying to develop code that can be used in other situations following the analyzed techniques.

Extreme programming is an approach associated with the agile methodologies for software developments. As suggested by the name, the approach is based on techniques that can be considered extreme in comparison to traditional programming development techniques. The approach was discussed by Hoffer, George, and Valacich (2011) with an explanation of the main characteristics that describe it. They include short-incremental development cycles, automated tests, and two-person programming teams. These techniques are part of the ideas that were implemented as part of the new approach. They were used as part of the supporting techniques for the language immersion approach mentioned. For example, the two-person programming teams were used in some initial programming exercises after introducing programs with important concepts.

The use of the techniques referenced, when combined effectively, can be associated with different theories and concepts that are important in education. Among others they include concepts associated with andragogy like the assumption related to the self-directed learning approach, theories associated with the collaborative learning approach, the experiential learning theories, the problem based learning approach, and other important ideas (Knowles, Holton, and Swanson, 2011; Merriam, Caffarella & Baumgartner, 2007; McKeachie and Svinicki, 2006; Palloff and Pratt, 2007; Wlodkowski, 2008).

Research Questions

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 1.1. What is the difference between students working with the proposed approach in terms of performance as compared to the performance of the students that received instruction using the traditional approach? (Quantitative Approach)

ResQ 1.2. What is the difference between the two groups of students on their motivation toward programming? (Quantitative Approach)

ResQ 1.3. What is the difference between both groups in the predisposition of students toward continuing in the program? (Quantitative Approach)

ResQ 1.4. How do the students feel in working with the proposed approach? (Qualitative Approach)

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 2.1. What is the improvement, if any, in the average time needed by students to get confident in programming using the main building blocks and structures required to be learned in the first programming class? (Quantitative Approach)

ResQ 2.2. Based on the Critical Incident Questionnaire (CIQ) instrument for formative assessment, how much do the computer science freshmen students participating in the study using the proposed approach to teaching programming benefit constantly from using it? (Qualitative Approach)

ResQ 2.3. How much time is needed to get confident with the different building blocks and structures while the understanding acquired according to the CIQ is improved using the proposed approach? (Mixed Approach)

ResQ 2.4. How much significant is the difference between the two groups of students in performance and how much positive is it in favor of the proposed approach? (Quantitative Approach)

ResQ 2.5. How much significant is the difference between the two groups of students in motivation toward programming and how much positive is it in favor of the proposed approach? (Quantitative Approach)

ResQ 2.6. How much significant is the difference between the two groups of students in the predisposition toward continuing in the program and how much positive is it in favor of the proposed approach? (Quantitative Approach)

Quantitative Hypotheses

$H_0$1.1: The means of the scores for the two groups are not significantly different.

H_A1.1: The means of the scores for the two groups are significantly different.

$H_0$1.2: Students using the proposed methodology were similarly motivated or less motivated toward using programming in comparison to those using the traditional approach.

H_A1.2: Students using the proposed methodology were more motivated toward using programming in comparison to those using the traditional approach.

$H_0$1.3: Students using the proposed approach were similarly predisposed or less predisposed to continue in the program in comparison to those using the traditional approach.

H_A1.3: Students using the proposed approach were more predisposed to continue in the program in comparison to those using the traditional approach.

$H_0$2.1: The learning of programming concepts for the students using the proposed approach will be similar to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

H_A2.1: The learning of programming concepts for the students using the proposed approach will be faster compared to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

$H_0$2.2: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly similar or negative about their understanding of the concepts presented.

H_A2.2: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly positive about their understanding of the concepts presented.

$H_0$2.3: The time needed to get confident with the different building blocks and structures will appear to be similar or higher and the understanding acquired as reflected in the CIQ will appear to be similar or lower while students learn with the proposed approach.

$H_A2.3$: The time needed to get confident with the different building blocks and structures will appear to be reduced and the understanding acquired as reflected in the CIQ will appear to be higher while students learn with the proposed approach.

$H_02.4$: The means of the test scores for the two groups are not significantly different or the mean for the group working with the proposed approach is significantly lower.

$H_A2.4$: The mean test scores for the group working with the proposed approach is significantly higher.

Research Design

Using a mixed approach the study was worked with two groups of randomly registered freshmen students in an introductory computer sciences course that received computer programming instruction using two different approaches. Students' performance from the groups was measured through the scores obtained from quizzes, and a final test (really implemented in pretest/posttest combination), and their experience was registered through the class with the CIQ instrument (formative assessment), at the end with two questionnaire instruments addressing motivation toward programming and predisposition to continue in the program, and a focus group.

Method Overview

The proposed study was developed using a mixed method approach in which quantitative and qualitative elements were managed. The main setup of the study was based on working with two groups of students enrolled in the first course of the Computer Sciences program in the university of the study. The name of the course is Algorithms and Programs Development I and it is offered in 60 hours throughout a normal semester. The groups were defined to be as homogeneous as possible with up to 20 students in each. One group was the experimental one that received instruction using the proposed approach. The other was the control group that received the instruction using the traditional approach.

According to Swanson and Holton (2005) the positivist approach, associated with the quantitative part of the study, establishes that "researchers generally seek out facts in terms of relationships among variables" (p. 19). In this case the main variables are the teaching approach used and the student's performance (final score). As described by Cooper and Schindler (2011) the relationship between these variables is asymmetrical that presents a situation of causation in which "A 'produces' B or A 'forces' B" (p. 151). The teaching approach represented by A (the independent variable), that is a categorical variable (innovative or traditional), produced an effect on student's performance represented by B (the dependent variable) associated with the final score that is a continuous or interval datum (numeric value). This situation is associated with the hypotheses $H_A1.1$ and $H_A2.4$ (research questions ResQ 1.1 and ResQ 2.4).

The two groups took a pretest and a posttest. The tests were the same for both groups since the material worked throughout the course was exactly the same. As stated, the difference between the two groups was in the teaching approach used for teaching the required material. Both groups were evaluated throughout the semester using quizzes, tests and assignments. Also classes were assessed using the Critical Incident Questionnaire (CIQ), a formative assessment instrument suggested by Brookfield (1995). Data for the study were acquired from the instruments mentioned: the tests, the quizzes, the assignments and the CIQ. Also summative data about the experience in classes were obtained from two different questionnaires and a focus group. The two questionnaire instruments that were used are the Modified Rosenberg Self-Esteem Scale and the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale. They were used previously with freshmen computer programming students in Bergin and Reilly (2006), and in Kranch (2010). A pilot study was not necessary because both instruments were tested and validated in previous studies with participants from the same population. The protocol for the focus group can be seen in the Appendix.

The data analysis was accomplished using Mann Withney U tests and *t*-Test in the quantitative portion of the study providing with answers to the hypotheses established. The other quantitative data emerged from the dates when the CIQ, the quizzes, the assignments and the tests were offered, and from the closed items included in the questionnaires. The qualitative data obtained from the focus group served the main purpose of analyzing the effectiveness of the proposed teaching approach providing the opportunity to confirm the results obtained from the quantitative portion of the study and from the CIQ.

Approach

The proposed study was developed using a mixed method approach. According to Johnson, Onwuegbuzie, and Turner (2007), "mixed methods research is, generally speaking, an approach to knowledge (theory and practice) that attempts to consider multiple viewpoints, perspectives, positions, and standpoints (always including the standpoints of qualitative and quantitative research)" (p. 113). Aligned with this definition some quantitative activities were worked in order to obtain appropriate data that can be helpful in producing concrete results capable of giving the necessary insight to answer the main research questions of the study. Concurrently, some qualitative activities were performed that helped in supporting the main facts obtained from the quantitative ones.

The fact of working these activities concurrently established that the mixed method research design had been worked using the triangulation approach. About this Swanson and

Holton (2005) wrote that "[t]he triangulation design collects both quantitative and qualitative data simultaneously so that the investigator can converge the data to make comparisons…" (p. 320). Warfield (2010) established "that research strategies that combined both types of quantitative and qualitative data and analysis gained the most from validity through triangulation, and from a more comprehensive illustrative description of the phenomenon" (Hurmerinta-Peltomäki and Nummela, 2006 as cited by Warfield, 2010, p. 32).

In the study, as described in Capella University's School of Business & Technology Dissertation Guide, the quantitative activities were worked through an experimental design and using the exploratory qualitative inquiry approach. The results obtained from the activities of both approaches were used to establish comparisons as part of the interpretation of the results obtained (Creswell & Plano Clark, 2010; Swanson & Holton, 2005; Warfield, 2010). The results from the qualitative activities served the main purpose of supporting those obtained from the quantitative activities.

Methodology Model

In the quantitative part of the study an experimental design was worked. The methodology model used within the experimental design is based on the pretest-posttest control group design as classified by Swanson and Holton (2005). As they mentioned, by using this design "the researcher can make comparisons of the posttreatment behavior between groups in addition to making a within-group pretest-posttest comparison for both groups" (p. 85). Referring to it as a pretest-posttest randomized experimental design, Trochim (2006) used a notation based on letters to represent it.

In the notation Trochim (2006) illustrated the experimental design based in the comparison of two groups using pretest – posttest notation. One group R is the experimental one with the treatment X, using pretest O and posttest O as the main data collection methods. The control group R have pretest O and posttest O but no treatment X. By representing both groups with R Trochim is addressing the importance of recognizing that both groups need to be homogeneous. The use of Os for the pretest and the posttest implies that they must be all based in the same examination instrument.

In the study, independent sample t-Test were planned originally to be used in comparing the final score results of both groups. However, because of the non parametric characteristics of the final sample, the Mann Whitney U test was used for this purpose. This is associated with the main dependent variable of the investigation, the students' performance. Also this strategy was used to compare the different times needed to get confident with the main building blocks and structures required to be learned in a first programming class. Finally the methodology was used to determine if there are differences in the motivation of students toward programming and their predisposition to continue in the program. In order to establish if internal validity is achieved, paired sample *t*-Test was used in comparing the pretest and posttest results for the experimental group. This helped in providing how much significant the difference between the pretest and posttest should be after the treatment for this group. Some correlations were also used to establish adequate relationships between items of the questionnaires.

The qualitative activities of the study were done following an exploratory qualitative inquiry approach. This was the one used in the study looking for participants' opinions and feelings about the approach used for teaching computer programming. Other qualitative major approaches resulted to be inappropriate for the characteristics of the study. The qualitative data were collected basically from different submissions of the CIQ instrument for formative assessment and from a focus group after course ending.

Rationale on temporal sequencing or order of elements

In the triangulation approach followed for the mixed method research done, both quantitative and qualitative activities occurred concurrently. The referred activities occurred isolated because there were not evident dependencies between most of them. As mentioned the study occurred in a real academic setup in which the two groups were enrolled in the first programming course of a computer sciences program. Students enrolled in both groups randomly, from the point of view of the study, according to their schedule preference. One of the groups was the experimental group and the other was the control group.

Initially both groups took the defined pretest that provided the first quantitative measure of the study. As established, only the experimental group pretest score was used in the paired sample t-Test executed with the corresponding posttest. Throughout the classes' progression different predetermined specific topics were assessed using the CIQ instrument. It provided qualitative data about the effectiveness in teaching the associated topic, according to students' opinions. Also, specific quizzes were administered after some specific topics providing a quantitative measure of the effectiveness of the process. From this data, time periods were determined and collected for comparison purposes between both groups using the Mann Whitney U test. In all cases the qualitative data helped in confirming the effectiveness reflected in the corresponding quizzes' scores. The times collected, as well as the results obtained in the quizzes, established the effectiveness of the proposed teaching approach, and the level of improvement over the traditional approach.

In the last class of the semester the Modified Rosenberg Self-Esteem Scale and the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale instruments were administered. Their results for both groups were compared using the most appropriate non parametric test found, the

Mann Whitney U test. Also some correlations between items within each instrument were computed to confirm the existing relationships using Pearson's r correlation coefficients. After the end of classes the posttest was administered to both groups in order to obtain the main results of the dependent variable, the final scores obtained for the class (a quantitative measure). Finally, a focus group was worked with participants from the experimental group. It provided qualitative data helpful in confirming the results obtained throughout the other described activities.

The logical and conceptual relationships between the different kinds of data and elements of the study are the following (see Figure 2):

- The pretest result of the experimental group was compared with the corresponding posttest result in order to confirm if internal validity is achieved. The paired sample *t*-Test was used with these results for the experimental group. The test will provide information about the significance of the difference between the pretest and posttest. These tests "indicate whether the difference is likely to be a "real" difference" (Swanson and Holton, 2005, p. 39). Trochim (2006), Hays (1973), and Sheffer, Barone, and Anders (2011) referred about the use of the paired sample *t*-Test in the context of pretest – posttest validity evaluation. Sheffer, Barone, and Anders used it formally in research analyzing the difference in level of training of nurses about treatment of tobacco use and dependency.
- The results obtained from the different quizzes and assignments administered in different instances throughout the semester provided information about the effectiveness of the teaching process and the times when the specific topics were successfully covered.

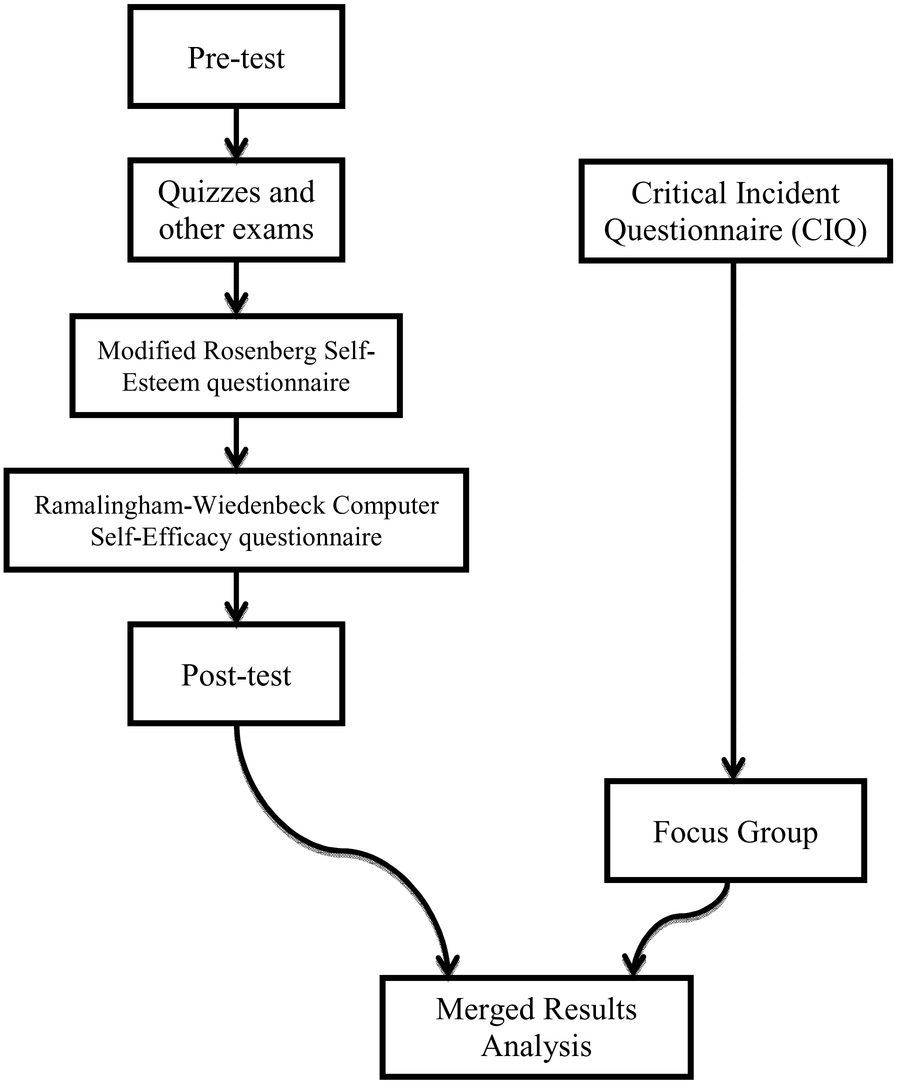

Figure 2. Mixed Method Design for the Study (Triangulation)

- The results of the CIQ instrument administered in different instances throughout the semester provided helpful information in confirming the results of effectiveness and times in weeks from the beginning of the semester obtained from the quizzes, assignments, and other exams.

- The time periods in weeks from the beginning of the semester obtained from the quizzes, assignments and other exams, and confirmed with the CIQ, from both groups were used to determine the relationship of effectiveness and improvement that may exist progressively between them. This were initially planned to be done using independent sample *t*-Tests between times corresponding to specific topics or learning areas but were finally done using Mann Whitney U tests because of the non parametric characteristic of the data. Trochim (2006) and Hays (1973) referred to independent sample *t*-Test as an appropriate statistic for comparison of mean values between two groups. Silverstone and Dadashova (2012) used this test in a research context comparing two groups of smokers submitted to different treatments in order to control the dependency to nicotine.
- The results obtained from the posttest in both groups, which correspond to the main dependent variable of the investigation, were evaluated to determine the relationship of effectiveness and improvement existing between them. As mentioned, Trochim (2006) and Hays (1973) referred to independent sample *t*-Test as an appropriate statistic for comparison of interval values between two groups, and it was used in this context by Silverstone and Dadashova (2012). However, due to the non parametric characteristics found in the data the test used were finally the Mann Whitney U.
- The results obtained from the Modified Rosenberg Self-Esteem Scale from both groups were evaluated to determine the relationship of similarity or difference existing in the motivation of students toward programming between them. Ritchey (2002) and Hernandez Sampieri, Fernandez Collado, and Baptista Lucio (2003) referred to independent sample *t*-Test as an appropriate statistic for comparison of

means in this context. This was evaluated based on the final circumstances of the study, and the Mann Whitney U test was the statistical procedure finally used, as described previously.

- The results obtained from the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale from both groups were evaluated to determine the relationship of similarity or difference existing in the predisposition of students toward continuing in the program between them. As mentioned Ritchey (2002) and Hernandez Sampieri et al. (2003) addressed the appropriateness of the *t*-Test in this context, and Fagerland, Sandvik, and Mowinckel (2011) described it as a powerful statistics in most situations but like the other statistics described the Mann Whitney U test was selected as more appropriate.
- The data obtained from the focus group were analyzed to confirm the results obtained from the quantitative results of the study applying the triangulation methodology as illustrated in Figure 2.

It can be established that the study followed the QUAN-qual model but using a full approach of triangulation that allowed the use of qualitative data collected to confirm or support the results obtained from the quantitative activities.

How the mixed design helped in answering the research question

Through the use of the triangulation method the research questions were answered combining the quantitative results in different aspects of the study with the qualitative data collected as information that helped to confirm those results. The triangulation approach was addressed clearly by Hernandez Sampieri et al. (2003), Swanson and Holton (2005), and Creswell and Plano Clark (2010). This implementation of the mixed method was evident in two main instances of the study. These were through the use of the CIQ (ResQ 2.2) and through the use of the focus group (ResQ 1.4).

The CIQ helped in confirming the effectiveness achieved in the learning process and the benefit obtained (ResQ 2.2) each time it was used in helping to support the results obtained from quizzes, assignments and tests (quantitative measures). While confirming this relationship the time period in which each topic was covered was determined (ResQ 2.3). Based on the specific topics, the times from both groups were compared (ResQ 2.1) in order to determine the general efficiency of the new approach to teach programming (ResQ 1) and the results were used in determining if it was better than the traditional approach (ResQ 2).

The focus group helped, like the CIQ, in confirming the effectiveness achieved in the learning process but in the whole course. The experiences lived by students in working with the approach used for the experimental group and their feelings toward it (ResQ 1.4) helped in supporting the results obtained in the posttest (the dependent variable of the study – ResQ 1.1, ResQ 2.4) and the other measures obtained from the instruments used (ResQ 1.2, ResQ 1.3, ResQ 2.5, ResQ 2.6). Those experiences allowed explaining and confirming the different results. Also this was important in determining the general efficiency of the new approach to teach programming (ResQ 1) and if the results were better than the traditional approach (ResQ 2).

Role of Researcher: "as instrument"

The researcher was involved in data collection management and analysis at different levels.

- Level 1: The involvement with quantitative results was in collecting the data, processing the adequate statistics, and analyzing the corresponding results. This applies to ResQ 1.1 and ResQ 2.4, ResQ 1.2 and ResQ 2.5, ResQ 1.3 and ResQ 2.6, and ResQ 2.3.

- Level 2: In the qualitative area the CIQ instrument results were read and analyzed by the researcher in reacting adequately in terms of classes' effectiveness, and analyzing critically about the effectiveness of the proposed approach during the course development. This applies to ResQ 2.1, ResQ 2.2 and ResQ 2.3. In terms of its use as a formative assessment tool, the results obtained were analyzed to make the appropriate adjustments in assuring the adequate understanding of the taught concepts.

- Level 3: In the qualitative area, the researcher was actively involved in the focus group exercise worked after course ending. In this activity the researcher worked doing an extra effort been a good listener, and analyzing critically the expressions done by students during the process. Part of this was the generation of new questions that provided additional information for the study. This applies to ResQ 1.4. This was worked with the target in mind of having the necessary information to confirm and describe the other results obtained in the study and to answer adequately the research questions.

Role of Researcher: Background Training & Experience

The experience of the researcher in collecting and managing the type of data used in the study is based on three important sources.

- The first is a broad experience in teaching programming courses, especially at introductory level. These courses were part of the regular program of the researcher as faculty from 1998 in the Computer Sciences Department. This provided the necessary expertise to work mainly with data collection methods associated with most of the quantitative activities of the study. These are the pretest, the posttest, the quizzes, the assignments and the other exams (ResQ 1.1 and ResQ 2.4, ResQ2.1 and ResQ 2.3). This was also important in working with the CIQ (a qualitative activity) because of its use as a tool for formative assessment (ResQ 2.2).

- The second is the work done as researcher for the Master of Sciences MSCE in Computer Engineering with a major in Software Engineering. The study applied a mixed method approach and it included a lot of processing of both quantitative and qualitative data. The statistics tools and analysis done in the quantitative activities of that MSCE study are very similar to those used in this study. This applies especially to the work associated with the administration and data gathering of questionnaire instruments (ResQ 1.2 and ResQ 2.5, and ResQ 1.3 and ResQ 2.6). The qualitative activities for that MSCE study were also similar in complexity to those done in the study. The difference was mainly in the methodology used but not in the gathering of the data and the skills used. The MSCE study was based in direct observations of activities and interviews done to all the participants. This study was based in the analysis and gathering of qualitative data obtained from the CIQ results (ResQ2.2 and ResQ2.3), and the focus group worked with some of the participants (ResQ 1.4). The experience acquired working with interviews was an important one in support of the need to facilitate the focus group component effectively.
- The third is the knowledge and skills developed as part of different learning processes in working with this type of data and the research process. These include trainings and experience developed through 14 years in the institution as faculty and as an

administrative staff in different positions (Academic Computing Chair, Computer Sciences Department Chair, and Assistant Dean of Academic Affairs), courses taken for professional improvement and at the master level, and the doctoral level courses taken at Capella University. The management and analysis skills necessary in working with assessment and the different evaluation instruments was acquired from the experiences as faculty, as department chair, and the multiple courses taken in education, including basic courses at the undergraduate level, one of College Teaching Practice at the master level, and the Post-Master's Certificate in College Teaching from Capella University (ResQ 1.2 and ResQ 2.5, ResQ 1.3 and ResQ 2.6, ResQ2.2, and ResQ2.3). The research skills in different contexts were acquired from the experiences as chair and assistant dean, from thesis worked in the master degree, and from the different courses taken in Capella University, including Survey of Applied Research Methods and Advanced Mixed Methods Research Designs (all research questions).

Role of Researcher: New Experience

For a researcher as with anyone there is always something new to learn, some skills that may be improved, and some doubts that need to be clarified. Some areas associated with the studies' design were new to the researcher or not fully under the optimal expertise. However, there were no problems with this situation because, through years of learning and experience in different contexts, the necessary knowledge and skills to address any situation in research was acquired. A strong level of experience helpful in the development of the study was already part of the researcher's background. From multiple sources of support available to the researcher, some were used when needed in clarifying doubts and improving procedures. The sources available included the support of the dissertation mentor; the support of faculty partners including researchers, statisticians, and faculty from computing and education areas; the multiple books and articles collected throughout years of experience and learning; and the resources available both in the Capella University's Library and the researcher's institution.

Sample

The Population

The population of the study is all the freshmen students taking an introductory course in computer programming in any computing program or any other program that requires introductory courses in computer programming. Examples of such computing programs are computer sciences, information systems, and software engineering. Examples of other programs that can have requirements in programming include some engineering programs, some science programs, and some business programs.

The Sample Frame and Sample

The sample frame is all the freshman students admitted to the Computer Science Program in the university where the study was conducted. The freshmen students admitted to the program are normally 90. In August 2012 they were 89.

The final list of students admitted was considered the final sample frame authorized by institution's administration to conduct the study. The students were all high school graduates with ages going from 17 to 19. They are normally good students that are admitted to the institution from the best in the country according to the admissions procedures established. In Puerto Rico the better students compete for the programs in the only one state university in the country that is compounded of eleven campuses.

The sample groups were originally intended to have up to 20 participants on each. This is directly associated with the maximum number of students that are allowed in each course section for COTI 3101 – Algorithms and Program Development I. This course is the first one in programming offered as part of a four year program in computer sciences.

Sampling Procedures

Within the context of the study having participants as described, in two groups of up to 20 freshmen students for each, their original selection resulted to be automatically random. This was because all students determined which course section they preferred to enroll in without any intervention from professors or any official from the institution. According to Plano Clark and Creswell (2008) this situation may be classified as in the category of cluster sampling that is part of the traditional probability sampling techniques. They established that this technique "occur when the sampling unit is not an individual but a group (cluster) that occurs naturally in the population" (p. 201). The groups in this case (the clusters) were selected as they existed in the first day of classes after students' decision of the section in which they wanted to take the course. In the enrollment procedure, students received a detailed explanation of the schedule available and the enrollment process, and they finally prepared their individual class schedule according to their preferences. The effect for the study was that participants were really randomly distributed for both groups.

The sections used in the study were previously assigned to the professors participating in the study without identifying the students as participants of a study. As a normal procedure these faculty members are assigned four months before and freshmen students are finally admitted three months before the first day of classes. The number of sections goes usually from five to seven with one defined for non-traditional students, that is offered in the night and another one that is open for other years' students that may need to take it, or existing students that come from other programs or institutions. The other sections are reserved for the incoming new students that select their schedule a month and a half before the first day of classes.

The students that were part of the focus group after class ending were selected randomly from the students participating as part of the experimental group. This focus group was intended to be worked with ten students. It is generally suggested for this type of exercise to have between eight and twelve participants. Because this sampling was of subjects from the same group with common characteristics and the purpose for selecting them was to obtain details about the phenomena under study, the sampling method used is known as homogenous sampling according to the description provided in the Capella University's School of Business & Technology Dissertation Guide.

Sample Size

As established, the sample size actually responded to the specific circumstances of the study. All course sections are of 20 students, and then two groups of up to 20 were intended to be used for the study in order to work adequately with the real setup of the institution.

As explained, there was not a recruitment process for the study since students decided the sections in which they enrolled for classes. The first contact with students was in the first day of classes. In the date determined as the first day of the study, that did not need to coincide with the first day of classes, when they were oriented about their participation in the study, before consenting, students had the opportunity to express concerns for their participation and they had the option to change to a different section of those not participating, or to decide staying in the section. From the section where the student opted to change, a volunteer was to be looked for participation in substitution of the one leaving the section of the study, if there was someone available. In any event, the study had the possibility to be worked with a smaller amount of participants than those originally proposed.

The active students' participation in the study occurred totally in the context of the course. They participated in regular classes, taking quizzes and exams, and using assessment instruments. Because of this, students not participating in the study were not affected. They were part of all the activities in the course. The last day of classes the participants answered two short surveys, and after taking the final exam, eight of the participants (from ten originally intended) with the proposed approach participated in a focus group. Students not participating in the study were not required to answer the surveys, and were not included in the random selection of the participants of the focus group. They were excused from being in the class during these activities that took about half hour.

As stated, the students that were part of the focus group were selected from those participating in the study randomly. This was done using a computer routine that generates numbers randomly. A random number was generated for each of all participants. The ten participants with the smaller numbers were the ten selected for the focus group, but only eight finally assisted.

Finally, the use of instruments by participants and the focus group activity occurred as follows:

- The CIQ was used in the last ten minutes of classes each week, when possible. This is an instrument normally used as a formative assessment tool.

- The two surveys were used in the last half hour of class in the semester.

- The focus group was worked after the final exam for the class. The final exam is normally assigned to occur a few days after the last class of the semester.

Rationale

From the research questions it is clearly established that two groups were needed. The results obtained from working with them were compared in determining the main issues associated with the study. Since the study was based on the context of groups of students in a real course of introductory computer programming concepts, the size of the groups was considered appropriate. All the groups that receive instruction in these type of courses in most of the institutions of higher education around the country use a similar setup for their courses. As a general rule the optimal number of students to have in classes in which computer are used for teaching is 20. This is the general rule used in the institution and for the study.

Setting

The study was conducted in a four year university institution. It was developed in the context of the first course offered to the freshmen students of the Computer Science Department. This setup was selected because it is the institution where the researcher works as professor. The context of the study required this as an optimal setup, because the proposed approach needed to be implemented by the researcher in order to avoid any bias if another professor was required to implement it without having a real understanding of the approach.

By working the study in the institution used the department and its students achieved a main benefit because the general strategy used in teaching computer programming was in formal evaluation. This provided possibilities of improvement, if the proposed approach resulted to be better than the traditional one. If the proposed approach resulted to be better, students may be directly benefited in the course treated and can be benefited in others that follow, as the strategy continues being implemented on them. The existing body of knowledge will be increased with a new approach to be considered by researchers in future studies, and other institutions with similar setups will have the opportunity to test its effectiveness for the benefit of their students.

Instrumentation/Measures

Data Collection Instruments

ResQ 1.1, ResQ 2.4 – Pretest, Posttest:

This instrument was the regular final evaluation exam used at the end of course in order to determine students' knowledge and skills at that moment. This instrument was developed in coordination between all faculty members offering the course. The modification of this instrument is normally focused on avoiding any misuse of previous versions by students who may be repeating the course or who may have close contact with students that already took the course. The instrument aims to establish if students have the necessary knowledge and skills in computer programming in order to design, develop, code, and test any type of algorithm with a reasonable level of effectiveness, but at an introductory level. The exam included both objective and subjective parts. The objective part included multiple choices, true or false, and filling the blank exercises. The subjective part included applied programming exercises.

For the study, the exam developed the last year was selected and analyzed in order to verify its appropriateness. Some modifications done to it were discussed with other professors offering the course. This was done prior to the beginning of the study because it was used in the pretest activity. All professors offering the course accepted to use this version of the exam but they really had the option to use a different one. The exam was developed for students taking the first course of computer science named Algorithms and Programs Development I in the context of the program offered in the study's institution. Most of the students taking this course were freshmen and this was totally true for the sections that were part of the study. Because of the characteristics described, there was not any psychometric information associated with this instrument. However, as part of the study the effectiveness of the exam in measuring students' performance was automatically demonstrated.

ResQ 1.2, ResQ 2.5 – Modified Rosenberg Self-Esteem Scale:

This is a ten item instrument designed to measure self-esteem or self-motivation that was adapted for computer programming by Bergin and Reilly (2006) as established by Kranch (2010). In its original version it is appropriate to measure self-esteem in any particular situation in diverse areas. In the modified version it is appropriate to measure self-esteem in working with computer programming. This established its appropriateness in assessing self-esteem or self-motivation toward programming in any study associated with the evaluation of computer programming knowledge and skills for both novice and expert programmers.

As established, an understanding of the programming process may be associated with motivating students toward this process in the context of the study. It was considered as evidence confirming the efficiency of the proposed approach throughout the evaluation of students' motivation toward the programming process at the end of the course. Bergin and Reilly (2006), Bergin and Reilly (2005), and Kranch (2010) described how the instrument was used in similar contexts. Bergin and Reilly (2006) established how the process of modifying the instrument for computer programming was worked. They also used it with 123 freshmen students of introductory programming courses from four institutions in the Republic of Ireland, and validated with a Cronbach's alpha value of .91.

Bosson, Swann, and Pennebaker (2000), and Schmitt and Allik (2005) referred to this instrument positively as a widely used one because it is short and simple. Schmitt and Allik argued about the instrument considering it psychometrically valid because of its demonstration of maintaining reliability and factor structure in a study worked in 53 nations. In the study the Cronbach alpha's mean among all nations was of .81 going from .45 in Congo to .90 in Israel and United Kingdom. Internal reliability among all nations was also verified "with an overall Guttman split-half reliability of .73" (p. 629). Other validity measures were also addressed in the referred study.

ResQ 1.3, ResQ 2.6 – Ramalingam-Wiedenbeck Computer Self-Efficacy Scale:

The 25-item instrument adapted for the study using the Java programming version developed from Askar and Davenport (2009) was originally created with 32 items for C++ programming according to Ramalingam and Wiedenbeck (1998). Kranch (2010) adapted and used a 26-item version also from the one developed by Askar and Devenport. The purpose of the instrument is to assess the computer self-efficacy of participants.

As established, the original version was developed for C++, but Askar and Davenport (2009) developed a version appropriate to assess Java programming self-efficacy. Self-efficacy is associated with the Bandura's Social Cognitive Theory developed in the 1970's (Bandura, 1977) and with the predisposition of participants toward continue acquiring specific knowledge or working with specific skills. This Java version was the one used for the study but adapted by deleting all the not applicable items. These are items that are not part of the discussion in the course used for the study.

In the context of the study this instrument was used to collect data regarding the predisposition of participants to continue working with computer programming. This was considered an indication that participants will or will not want to continue studying in the academic program. With the results obtained in the context of the study it was used to assume that students will or will not be predisposed to continue in the program.

The instrument was validated in its original version as described by Ramalingam and Wiedenbeck (1998) with 421 students of an introductory C++ programming course. Its overall reliability was calculated with a Cronbach's alpha of .98. In the Java programming version Askar and Davenport (2009) demonstrated the instrument's validity and reliability with 326 students with a Cronbach's alpha score of .99.

Data Collection

Units of Analysis

The study consisted of two groups. The control group received instruction using the traditional approach for teaching computer programming. The experimental group received instruction using the proposed approach. This established the main independent variable of the study. The dependent variable was the students' final performance expressed in the scores obtained in the posttest for both groups. This variable established the main unit for analysis in order to determine two main things: whether the proposed approach was shown to be efficient, and whether this new approached was demonstrated to be better in teaching computer programming to freshmen computer sciences students. Improvement of the teaching approach had been demonstrated if both of the described situations occur, supported in the analysis of data from the other quantitative and qualitative sources.

Constructs, Phenomena, Issues, or Elements of Interest

These were the main constructs of the study:

- Course group (the main independent variable)
- Students' performance in the posttest (the main dependent variable)
- Students' performance in the pretest
- Students' performance in quizzes

- Students' responses to the CIQ instrument
- Students' results from the Modified Rosenberg Self-Esteem Scale
- Students' results from the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale
- Students' responses in the focus group

Conceptual Definitions

The conceptual definitions of the constructs in the study were the following:

- Course group (the main independent variable)
  - The course group construct was the main independent variable of the study. The control group received the instruction of computer programming using the traditional teaching approach. The experimental group received the instruction using the proposed approach.
- Students' performance in the posttest (the main dependent variable)
  - This construct was measured through the final score obtained in the posttest. This was the final test that students took in demonstrating the whole knowledge and skills acquired throughout the semester.
- Students' performance in the pretest
  - The pretest was exactly the same instrument used for the posttest. It was administered in the first days of the course in order to verify the amount of knowledge and skills that students had about the topics of the course. It was expected that these scores results were low and very similar for all students in both groups.
- Students' performance in quizzes, partial tests and assignments

- The quizzes, partial tests and assignments offered after working with specific topics throughout the semester provided scores that were helpful in determining the progressive improvement in performance. However, they also provided the times needed for comparison between both groups in determining the differences occurred among the corresponding topics. The idea was to determine if some specific main building blocks or structures in the course were learned successfully in a shorter time in the experimental group when compared to the control group.

- Times when main building blocks and structures are covered
  - These times will be determined based on the offering of quizzes, partial tests, assignments and the completion of the CIQ that confirm the learning of the main building blocks and structures covered in the course. These measures provided for part of the evaluation of the effectiveness and possible improvement provided by the proposed approach. Through this the effectiveness of the immersion language theory and the use of extreme programming were evaluated.
- Students' responses to the CIQ instrument
  - The CIQ is a broadly used formative assessment instrument suggested by Brookfield (1995), Brookfield (2006), and Wlodkowski (2008). It provided for the qualitative evaluation of the topics covered by students in identifying if the teaching process had worked effectively in providing knowledge and skills. This was used to support the results obtained from the quizzes, partial tests and assignments in determining the times needed for comparison

between both groups for comparison of the differences occurred among the corresponding topics, as described.

- Students' results from the Modified Rosenberg Self-Esteem Scale
    - A good understanding of the teaching-learning process as a successful one in the context of the study was associated with an improvement in motivation of students toward programming. This measure was considered as evidence of the efficiency of the resulting learning process and the motivation of students toward programming at the end of the course. In this context it was used by Kranch (2010) working with a version of the instrument modified and validated for computer programming as described in Bergin and Reilly (2005), and Bergin and Reilly (2006). In Bosson, Swann, and Pennebaker (2000) the original Rosenberg Self-Esteem Scale (RSES) was described as "a 10-item scale that measures people's feelings of global self-worth" (p. 633). Schmitt and Allik (2005), citing other researchers, described it in the following terms (p. 623):

        > Among the many devices for assessing global selfesteem, the self-report version of the Rosenberg Self-Esteem Scale (RSES; Rosenberg, 1965) remains the most widely used measure (Byrne, 1996; Wylie, 1974). The popularity of the 10-item RSES has been due, in part, to its long history of use, its uncomplicated language, and its brevity (it takes only 1 or 2 min to complete).

    - Schmitt and Allik (2005) concluded through a study addressing the effectiveness of self-esteem instruments within 53 nations of diverse cultures that the RSES was demonstrated to psychometrically maintain their reliability and factor structure.

  - Bergin and Reilly (2005), and Bergin and Reilly (2006) described how they modified this widely used instrument adapting it for evaluation of motivation toward Java computer programming. They tested the instrument with 123 freshmen students of introductory programming courses from four institutions in the Republic of Ireland.

- Students' results from the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale
  - The motivation of students can have important implications in their attitude toward working with programming, as established by Bergin and Reilly (2005). This also occurs with their confidence about the competence they have to develop programs efficiently. It was considered important to analyze the efficiency achieved in the context of the study and in the general context of the self-efficacy theory, as applied in this case to the use of computers to develop programs. However, there was also a very important implication associated with the main purposes of the study. It was that exists a better probability that motivated and self-confidence students opt to continue enrolled in the academic program. This promotes improvement of retention rates and implies the students' better understanding of computer programming concepts.
  - Kranch (2010) successfully used this instrument in a study context that was very similar to the one of the current study, working with novice students. However, from its creation in 1998, the instrument was validated, as

Ramalingam and Wiedenbeck (1998) described, with 421 students of an introductory C++ programming course. Recently, Askar and Davenport (2009) demonstrated the instrument's validity and reliability for a Java version tested with 326 freshmen students from different engineering and science disciplines that requires computer programming. This Java version was the one used in Kranch (2010) and the one to be used in the study after adapting it appropriately. Askar and Davenport also mentioned the relationship existing between the instrument and the Bandura's Social Cognitive Theory developed in the 1970's that is generally known as the self-efficacy theory.

- Students' responses in the focus group (see the protocol in Appendix)
  - In the context of the study it was important to know how comfortable the students were in working with the proposed approach, giving information that was important to explain the results obtained quantitatively. This was the main purpose of the focus group in this context. This technique is widely used in research and education to obtain information about the experiences lived by participants in a variety of situations and contexts.

Variables (Definitions of Constructs as variables) (QUAN)

- Independent variable
  - Course group
- Dependent variable
  - Student performance (posttest score)
- Other dependent variables
  - Pretest score

- Quizzes, partial tests and assignments scores
- Times when main building blocks and structures are covered
- Modified Rosenberg Self-Esteem Likert Scale results
  - This instrument was used in studies with participants from the same population in the works described by Bergin and Reilly (2006), and Kranch (2010).
- Ramalingham-Wiedenbeck Computer Self-Efficacy Likert Scale results
  - This instrument was used in studies with participants from the same population in the works described by Askar and Davenport (2009), Bergin and Reilly (2006), and Kranch (2010).

Observational Definitions (QUAL)

- CIQ assessment instrument results
  - This is a formative assessment instrument discussed by Brookfield (1995), Brookfield (2006), and Wlodkowski (2008) to work effectively in evaluating the effectiveness of the teaching process.
- Focus group results
  - Through the focus group information about the feelings of students working with the proposed approach was collected in order to determine how comfortable it was for them and how much their expressions supported the results obtained from the quantitative part of the study.

Operational Definitions

- The main dependent variable, student's performance, was measured using the posttest end of course score result from all students in both groups. This score followed the standard percent scale used for all courses exams (*0%-100%*) but came from a raw score of 80. Scores of over 70% are considered normally approved for students in the context of the rules established by the Computer Sciences Department in the study's institution. For the study an independent sample *t*-Test were originally intended to be used as the statistical test to measure significance in comparing the resulting raw scores obtained from the two groups. In the case of having non parametric characteristics, as occurred, the Mann Whitney U test was considered a more appropriate statistics in the described situation.

- Pretest score was measured through a beginning of course exam result obtained from all students in both groups. This score followed the standard percent scale used for all courses exams (*0%-100%*) but came from a raw score of 80. Scores of over 70% are considered normally approved for students in the context of the rules established by the Computer Sciences Department in the study's institution. For the study a comparison of these score was not required according to the design proposed.
- Students' performance in quizzes and other instruments was measured using scores based on 40 points in most cases but standardized to the 100% scale. This allowed simplifying the consideration of having over 70% as the base for approval according to the rules established by the Computer Sciences Department in the study's institution. For the study an independent sample *t*-Test were originally considered to be used as the statistical test to measure significance in comparing the resulting raw scores obtained from the two groups in some quizzes and other activities measuring the same knowledge and skills. In the case of having non parametric characteristics, as occurred, the Mann Whitney U test was considered a more appropriate statistics in

the described situation. This was considered to confirm tendencies identified with the most important measures of the study. However, it was found that this measure was not totally homogeneous and was also of less importance in the context of the research questions of the study.

- The times when main building blocks and structures were covered were measure in weeks after determining the date of offering of the corresponding quizzes, partial tests or assignments, and their validation with the results from the corresponding expressions obtained from the CIQ instrument. The times obtained were used for comparative purposes with the corresponding times from the other group. For example, if a quiz and the CIQ was about the knowledge of the "for" statement, the time about it was used for comparison between the two groups in order to determine which one occurred first. The times from both groups were used to establish if there is a general difference between groups, after extracting any times corresponding to topics or areas that were not discussed to the students of either of them, and those that occurred simultaneously. An independent sample *t*-Test was intended to be performed with the times obtained in order to determine this. The Mann Whitney U test was used because of the non parametric characteristics found in comparing the data.
- The Modified Rosenberg Self-Esteem Scale results were obtained by completion of the instrument by participants. The instrument is based on a four point scale with ten items. It was adapted and tested for computer programming by Bergin and Reilly (2006) and used later by Kranch (2010). Both of them worked with the instrument in the context of first year computer programming students. As defined, it is based in

having -2 for strongly disagree, -1 for disagree, 1 for agree, and 2 for strongly agree. However, for the study a simpler coding was worked for statistical purposes going from 1 for strongly disagree to 4 for strongly agree. Possibly looking for reducing response bias, the instrument had been defined using negative statements that were coded in the reverse order. For the study this was considered carefully in working with the results. An independent sample *t*-Test was originally considered to be used as the statistical test to measure significance in comparing the resulting mean values obtained from the two groups. This statistic is appropriate in these cases as described by Ritchey (2002) and Hernandez Sampieri et al. (2003). However, due to the final characteristics of the data the test used was the Mann Whitney U.

- The Ramalingham-Wiedenbeck Computer Self-Efficacy Scale results were obtained after completion of the instrument by participants. This instrument is based in a seven point Likert scale going from "not at all confident" (1) to "absolutely confident (7). Kranch (2010) used an adapted version for Java programming with 26 items but it was originally created for C++ as described in Ramalingam and Wiedenbeck (1998) with 32 items. The version used in the study and the one used finally by Kranch was the one adapted and validated for Java by Askar and Davenport (2009) with 326 freshmen students from engineering and science programs. An independent sample *t*-Test was considered to be used as the statistical test to measure significance in comparing the resulting mean values obtained from the two groups, if parametric assumptions are met after testing the data. This statistic is appropriate in these cases as described by Ritchey (2002) and Hernandez Sampieri et al. (2003). However, since parametric assumptions were not met the test finally used was the Mann Whitney U.

Relationships among Variables

- The students' performance (dependent variable) was believed to be related to the course group (independent variable) in a cause and effect relationship.
- The students' performance was believed to be related with the Modified Rosenberg Self-Esteem Scale results. The self-esteem results supported the obtained scores reflecting the students' performance.
- The students' performance was believed to be related with the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale results. The self-efficacy results supported the obtained scores reflecting the students' performance.
- The pretest score from the experimental group was compared to the students' performance (posttest score) in order to verify for internal validity using paired sample *t*-Test.

Rationale

As established, the dependent variable of the study was the students' final performance expressed in the score obtained in the posttest for both groups, and it was the one identified as the main unit for analysis of the phenomena under study. The comparison between the scores obtained from both groups established two main things. The first was whether the proposed approach resulted in being efficient in providing the knowledge and skills required. The other was whether the proposed approach resulted in being better than the traditional approach for teaching in the context of the study. The focus group directly served the purpose of supporting the results obtained from students' performance with students' opinions about the proposed approach and how comfortable they felt with it.

Data Collection Methods

For ResQ 1.1 and ResQ 2.4:

After COTI 3101 course ending all students were cited to take the final exam. The date and location were established formally by the registry office as normally occurs. All sections took the exam together in the same room. The day of the exam all students were cited to begin at a specific time. Students proceeded sitting where they preferred but assuring to leave an empty seat between any of them and other students. They were instructed to do so when they entered to the exam location. After all students were seated the exams were distributed by each faculty member to their corresponding students. The students participating in the study in both the control and the experimental group received the same exam. They were supposed to be unaware about the concept of being part of a specific group from the study.

After exam instruments were distributed some general instructions were imparted by one of the faculty members about important issues, the time length available to answer the exam, the specific procedure to obtain clarifications for any doubts, and to return the answered exams. These instructions included the need to be in total silence during the exam period, indications about the procedure of requesting any help from a faculty member and of getting permission to go out of the location, the need to return the answered exams to the professor who taught them the class, the availability of two hours to answer the exam, and a general explanation of what was expected from working the problems included in the practical part of the exam. After this process they received the indication to begin. The students that arrived late to the exam were seated according to the available spaces and received some individual instructions. After each student finished answering the exam he or she returned it to his or her professor and was instructed to sign a sheet indicating that the exam was returned answered.

In the following days the exams were corrected by the faculty members. The objective part was corrected by the faculty member who taught the class to each group. However, the practical part was based on predefined rubrics prepared for each problem included. The total score of the exam, based on 80 points, was determined after correcting each exam, and the final percentage was calculated accordingly by dividing the obtained total by 80 and then multiplying the result by 100. The obtained percent value was register by the professor in the class roll book and by the researcher in Excel and SPSS documents next to the number of each participant. After all values were registered for all participants in each group a statistical test was computed with the data for the control and the experimental groups comparing the difference between them and how significant the difference was. Initially the test considered for this was the independent sample *t*-Test but after verifying the non parametric characteristics of the results the Mann Whitney U test was used. This test was extensively discussed in Fay and Proschan (2010).

For ResQ 1.2 and ResQ 2.5:

In the last class of the semester both the control and the experimental groups students were instructed to answer two questionnaires as part of their participation in the study. The first was the Modified Rosenberg Self-Esteem Scale questionnaire. Students were not aware about which group they belong to, but they knew they were participating in a study as consented by them in the beginning of the semester. Before answering the questionnaire students were instructed to answer it sincerely, to be aware about the meaning of the scale to be used, and that they were going to have ten minutes to answer it. This was considering that the instrument had only ten items. After they received the instructions the questionnaire was distributed immediately. In ten minutes the questionnaires were going to be collected by the professor.

When the class was finished the researcher worked with the tabulation of the answered questionnaires. The answers obtained from both groups were registered in SPSS. After all values were registered for all participants in each group a statistical test was performed with the data for the control and the experimental groups in comparing the difference between them and how significant the difference was.

For ResQ 1.3 and ResQ 2.6:

As indicated, in the last class of the semester both the control and the experimental groups students were instructed to answer two questionnaires as part of their participation in the study. The second one was the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale questionnaire. Students were not aware about which group they belong to, but they knew they were participating in a study as consented by them in the beginning of the semester. Before answering the questionnaire students were instructed to answer it sincerely, to be aware about the meaning of the scale to be use, and that they had 30 minutes to answer it. This was considering that the instrument had 25 items. After they received the instructions the questionnaire was distributed immediately. After 30 minutes the questionnaires were collected by the professor.

When the class finished the researcher worked with the tabulation of the answered questionnaires. The answers obtained from both groups were registered in SPSS. After all values were registered for all participants in each group a statistical test was performed with the results obtained through the program from the data for the control and the experimental groups in comparing the difference between them and how significant the difference was.

For ResQ 1.4:

Following a random sampling procedure, ten students were selected from those participating in the experimental group in the last class of the semester. The ten selected students were immediately notified and requested to attend a meeting that occurred after the final exam. These students were oriented about the purpose of the meeting, clarifying that it did not have any relationship with their performance in the class, and that it did not have any effect on the final grade. They were notified that the meeting was strictly part of the study.

On the day of the focus group, the meeting room was prepared with some appetizers, coffee and juice for the participants before beginning the process. A sign was placed on the door indicating that a meeting was in progress and that no disturbance was allowed. When students arrived they were received by the researcher inviting them to eat and drink something, and to feel comfortable in determining where to sit. After students arrived 15 minutes were given for them to eat, drink and accommodate. Only eight participants finally arrived. The beginning of the focus group was then announced inviting them to sit.

The process was initiated explaining to students the purpose of the meeting and that they can feel free to express or ask anything about the approach used in teaching the class and the issues under discussion. They were also oriented about their right to leave the meeting at any moment if they wanted or if they felt uncomfortable. None of them left the meeting. A mention had been done about the length of the meeting too, and that it was going to be recorded to capture exactly all the discussions. It was planned for an hour but with the option to be extended for an extra half hour if students felt they needed more time to make their contributions. This situation occurred and the process continued after an open discussion at the end of the first hour with an agreement to do so.

The focus group began with initial guiding questions from the researcher. The discussion continued according to the different experiences discussed by participants. During the process new questions were integrated accordingly. The process continued until the time had gone after an hour and a half as indicated. At the end of the process the researcher thanked all students for their participation in the meeting and invited them to eat and drink something more. Also they had the opportunity to ask questions about the meeting or about the study. The meeting was formally finished allowing them to leave the room as desired.

After the focus group all the recorded expressions were heard and analyzed by the researcher. The data generated was analyzed by topics and registered in computer for easier management.

For ResQ 2.1:

In the normal development process of the course under analysis, the times when different building blocks and structures required for programming had been completed were determined in order to compare if any differences between the control and the experimental group. Based on the data obtained from different sources, the time when each specific area had been completed was determined. One of the sources was the CIQ in which students expressed weekly their feelings about their individual learning process. As part of this the topics in discussion were mentioned. This allowed tracking when specific areas of discussion were completed helping to obtain the amount of weeks passed from the first one of the course.

The other source of data was the quizzes, assignments and other exams presented throughout the semester. After analyzing the results of using these tools, the dates when they were presented helped also in determining the time when specific areas were completed in weeks from the beginning of the course. The data and the information obtained from these two sources and from teachers were analyzed in order to determine a single time of completion for each main building block or structure completed. This information was registered in Microsoft Word, Microsoft Excel and SPSS for the appropriate analysis.

After both classes ended, all the times registered for some important areas covered in the two courses were used to compute an average for each group. Any area covered only in one of the two groups was not included in the average computation. Areas identified that were covered simultaneously in both groups are also not included in the computation. Only the data that showed if any difference between the two groups existed was included.

For ResQ 2.2:

The CIQ is a formative assessment instrument that was used periodically with the main purpose of determining students' strengths and weakness associated with the learning experiences lived throughout a period. Students were instructed in the last ten minutes of the selected class to access the CIQ through the class web page in Moodle and to answer the five questions included briefly. In the beginning of the course the use of the CIQ throughout the semester was explained to students in order to help them in understanding clearly its use. This included the role of students and of the professor in using the instrument. They were urged to answer the questions sincerely because this would be helpful to the professor in making adjustments in the following classes and promoting a better understanding of the topics under discussion. The anonymity of each student's answers was granted indicating that the professor will only have access to the answers provided to each question without having any association with the students that answered them.

Aside from the use of the CIQ as an assessment instrument for the study, the answers were analyzed to determine the effectiveness of the proposed approach used for teaching programming. In other words, any benefit obtained by the students in working with the proposed approach was corroborated with the answers provided by them. The data obtained each period were registered in Microsoft Excel, including the week number, the topics or areas worked in the period, and the answers provided by all participants. From the registered data a general conclusion was developed based on the frequency achieved of situations described as having been or not been of benefit for students in learning using the proposed approach.

For ResQ 2.3:

As described for ResQ 2.1, through combining data from the CIQ and from the results of other tools the times in which specific building blocks or structures had been completed were obtained. The times obtained associated with each main building block or structure were registered in Microsoft Excel in order to have an idea of how much time was needed to get confident in working with each one. In other words, the times described when the specific area or topic was considered to be already learned.

Rationale

The data collection methods used were based mostly on working with specific values that describe straightforwardly the answers of all the quantitative based research questions. This is true for the use of the pretest, the posttest, the quizzes, assignments and other exams, the Modified Rosenberg Self-Esteem instrument, and the Ramalingham-Wiedenbeck Computer Self-Efficacy instrument. All of them produced single averaged values that can be used easily in analyses or for comparative processes, combined with the appropriate statistics, which will be helpful in answering the research questions.

Treatment/Intervention

In the experimental group the special treatment implied only to use an approach for teaching different from the traditionally used. Different strategies used from the traditional approach included:

- Students were directed to analyze the different code statements from a program trying to identify their use and functionality, and what happen with them in the context of the programs. The idea was that the explanation of the purpose of the different statements and building blocks emerge from students and not from the professor. This was referred as programming language immersion.
- Changes were also implemented in the specific order in which the different topics were normally presented, including addressing more complex topics earlier in the semester. This was referred as programming topic immersion.
- Extreme programming (XP) techniques were used in which students worked in pairs (pair programming), implemented solutions in short time, and tested the solution at all stages of the development process.
- Collaborative learning was used intensively at different levels and times. It occurred at the whole group level, in groups of four to five students, and in pairs.
- In support of the referred interventions, the teaching process was worked promoting the benefits of experiential learning and transformational learning. In this, problem based learning was used as an essentially important tool for these type of courses. As part of this, students worked also in exercises that were expected to promote in them to be self-directed learners.

- The use of the Critical Incident Questionnaire (CIQ) as a tool for formative assessment was of excellent support to the teaching learning process with the proposed approach. However, it was used similarly in the control group as a strategy for implementation in all courses. For teachers giving sections that were not part of the study, the use of this instrument remained to be optional. In the context of the study it was also used as a data collection method.

Data Analysis

Type of Data

Table 4. Types of Data from Different Quantitative Constructs

Hypothesis #	Dependent Variable Name	Data type (nominal, ordinal, interval, or ratio)	Independent Variable Name	Data type (categorical, ordinal, interval, or ratio)
H1.1, 2.4	Posttest Score	Interval	Course Group	Categorical
H1.2, 2.5	Modified Rosenberg Self-Esteem Scale Score	Ordinal	Course Group	Categorical
H1.3, 2.6	Ramalingham-Wiedenbeck Computer Self-Efficacy Scale Score	Ordinal	Course Group	Categorical
H2.1	Times when main building blocks and structures were covered	Interval	Course Group	Categorical

Statistical Analysis

H1.1, H2.4: From the final scores obtained after the posttest submissions to the students of both the control and the experimental group, the statistics that were originally intended to be used for comparison was the independent sample t-Test. This statistic establishes a measure of comparison between the two groups based on the means of the values obtained from each of them. It provides also information about how significant the difference between the two groups is. In the study, if a difference existed between the two groups and the scores of the group working with the proposed approach were higher, it can be inferred that its use is an important factor in improving student's performance. This is a parametric test that can be used if the samples were independent, they were assumed to be normally distributed and had equal variances with the measurement scales as intervals (final scores), according to Cooper and Schindler (2011).

However, the situation was different because of the non parametric characteristics finally found on the data. The samples were independent, but the groups were not normally distributed and did not have equal variances. The SPSS statistical software provides a non parametric test adequate for these circumstances known as the Mann Whitney U test. The appropriateness of the test in such circumstances was broadly discussed in Fay and Proschan (2010). Thus, the Mann Whitney U test was the statistical procedure used for all the comparisons needed between the experimental and the control groups.

H1.2: The Mann Whitney U test was used also in comparing the motivation of students toward programming between both groups by mean of the Modified Rosenberg Self-Esteem scale. This instrument's ordinal based results were treated after processing with SPSS using the Mann Whitney U test statistics as explained previously according to the explanations presented in Fay and Proschan (2010). For the study, if a significant difference was established between both groups and the mean of the experimental group was higher it can be argued that the proposed approach had positive implications in the motivation of students toward programming.

H1.3: With the Ramalingham-Wiedenbeck Computer Self-Efficacy scale the same situation occurs. This instrument's ordinal based results were treated using the Mann Whitney U test statistics. If a significant difference was established between the two groups and the mean of the experimental group was higher it can be argued that the proposed approach had positive implications in the predisposition of students to continue in the program.

H2.1: The time averages obtained for each student of both groups received a similar treatment. The Mann Whitney U test statistic was used to compare the means of the averages obtained for the students of both groups. The averages were interval values too. If a significant difference was established between the means of the time averages of the two groups, and the mean of the experimental group was lower, it can be argued that the proposed approach produced a reduction in the times needed to learn completely the main building blocks and structures used in the first computer programming course.

Qualitative Analysis

The qualitative activities of the study were done following an exploratory qualitative inquiry approach. This was the one used in the study looking for participants' opinions and feelings about the approach used for teaching computer programming. Other qualitative major approaches like phenomenology research or ethnography research resulted to be inappropriate for the characteristics of the study. The qualitative data was collected basically from different submissions of the CIQ instrument for formative assessment and from a focus group after course ending.

As established, the qualitative activities of the study were done following an exploratory qualitative inquiry approach, and the qualitative data will be collected basically from different submissions of the CIQ instrument for formative assessment and from a focus group after course ending. The reasons for thinking these data collection methods were appropriate for the research questions of the study, and consistent with the exploratory qualitative inquiry approach and the triangulation method were the following:

- Because the purpose of both the CIQ and the focus group was to look for participants' opinions and feelings about the approach used for teaching computer programming. This is consistent with the exploratory qualitative inquiry approach as established in the Capella University's School of Business & Technology Dissertation Guide.
- Because the CIQ allowed for information gathering on a regular scheduled basis about students' feelings associated with the teaching approach that helped in determining if it was of benefit for them in the classes (ResQ 2.2).
- Because the CIQ allowed for information gathering on a regular scheduled basis about students' understanding of the main building blocks and structures, helping in determining through the times associated with the periods when each area was completed, if the proposed approach improved the teaching of computer programming (ResQ 2.3).
- Because the focus group provided information of the feelings and opinions of students about the proposed approach, helping to determine if they felt comfortable in working with it (ResQ 1.4).
- Because the data obtained from both the CIQ and the focus group were helpful in drawing general supportive conclusions of the results obtained from the quantitative activities (triangulation method).

Types of Data

The types of data associated with the data collection methods for the qualitative activities were described in Table 4.

Table 5. Types of Data from Different Qualitative Collection Methods

Data Collection Method	Type of Data
CIQ …..…………………..	Verbal data in handwritten form on paper or typed form through a web page
Focus Group ...………..	Sound data transformed to verbal data in transcript form, typed

Data Preparation

For the CIQ:

- Students responded the CIQ through a web page as indicated by the professor. This exercise occurred periodically (every one or two weeks) for ten minutes at the end of the last class in the period.
- All entered data was accessed and analyzed after each submission in order to identify completed or in progress topics.
- Topics status and the corresponding dates were registered in Microsoft Word.
- Additional annotations were made after identification of areas as fully completed, including the week of completion.
- Opinions and feelings identified were registered within categories identified in a separate table as allowed by the software.
- Each category was identified for any further reference.

For the Focus Group:

- The recorded conversation (raw data) was heard and important annotations were made using Microsoft Word.

- The recordings were copied and stored separately for security purposes. One of them was a backup copy stored securely.
- Opinions and feelings identified were registered within categories identified in a separate table as needed.
- Each category was identified for further reference.

Data Analysis

For the CIQ:

The analysis of the data collected through the CIQ was centered in identifying the main area of discussion, an understanding of the instrument, and the completion of the instrument. To determine if the area was completed the data from different weeks were compared to establish the week when this really occurred. For example, if the topic associated with nested selection statements was considered one of the main areas it can occur that its discussion began in the first class of the fourth week but it may not be ended until the first class of the fifth week. This topic appeared commented in the CIQs of both the fourth and fifth weeks, but it did not appear reflected in the sixth one. This indicated that the topic was completed in the fifth week.

The expressions made in the CIQ were also analyzed in determining opinions and feelings of students about the course during the period. If a comment was consistent among the majority of the students about misunderstanding of the variable concept and they indicated that it was not clearly explained, the opinions associated were registered adequately in any appropriate category related with the situation, and how this was related with the proposed approach.

For the Focus Group:

Through the focus group the opinions and feelings of students about the proposed approach were the center of the analysis in order to determine if they resulted beneficial for them. All the expressions were analyzed looking for patterns that were helpful in providing for the formulation of strong conclusions.

Data Presentation

For the final presentation of qualitative data all of them were included in tables. Some charts were included for improved understanding of some ideas. Some important expressions done throughout the focus group were included as needed and as part of the results. Finally, all the findings from these data were discussed and associated with the findings obtained from the quantitative activities.

Risk Level Estimate for Qualitative Data

- Initial contact(s): minimal

  The initial contact with the final participants occurred the first day of classes.

- Sample recruitment & consenting procedure(s): more than minimal

  The random sampling occurred automatically as part of the standard procedure established for the enrollment of freshmen students.

- Preliminary meetings (e.g., training) (if applicable): more than minimal

  The only preliminary meeting that occurred was part of the initial orientation provided by the Computer Science department. In it all students were informed about the occurrence of the study throughout the first classes with some of them.

- Data collection procedure(s) (including 'member checking,' etc.): minimal

  The data collection procedures occurred totally hidden to students, except for the focus group. However, in the focus group their anonymity or confidentiality in the process were totally maintained during analysis and data management procedures.

- Data management procedure(s) (confidentiality breach): more than minimal

As specified, the anonymity or confidentiality of students were totally maintained during these procedures.

- Data analysis procedure(s) (confidentiality breach): more than minimal

  It was totally unnecessary to expose any information about participants in the analysis procedures due to the characteristics of the study.

- Write-up/presentation (confidentiality breach): more than minimal

  Similarly in the write-up and presentation of results and the related processes it was totally unnecessary to expose any information about participants.

Analysis Integration

As established, the triangulation method was used in describing the integration of quantitative and qualitative data to draw conclusions associated with the main research questions of the study. Both quantitative and qualitative data were collected throughout the development of the study. This was helpful in establishing ideas that allowed integrating the results of both areas in supporting the main conclusions of the study.

For ResQ 1 the quantitative results were obtained from a final exam described as the posttest, the survey about students' motivation toward programming described as a self-esteem instrument modified for computer programming, and the survey about students' predisposition to continue in the program described as a self-efficacy instrument about computer programming. The qualitative results were obtained mainly from the focus group and were confirmed with the experiential data previously obtained from the CIQ use throughout the semester. All of these results were considered in determining the efficiency of using the proposed approach for teaching computer programming to computer sciences freshmen students.

Similarly all of the quantitative and qualitative results described were helpful in answering the ResQ 2 research question. However, additional elements were added that strongly helped in determining how the proposed approach provided better results for teaching computer programming to computer sciences freshmen students. In terms of quantitative results the additional elements included the results obtained from quizzes, assignments and other exams presented throughout the semester, and the times determined when each main building block or structures was completed, that were values at the same time determined from the CIQ results, and the quizzes, assignment and other exams presented. The additional qualitative element was based also in the CIQ and helped in determining if there was a recurring benefit from working with the proposed approach.

Validity and Reliability

The Critical Incident Questionnaire (CIQ) is an instrument developed to be used as a tool for formative assessment. It was widely discussed in Brookfield (1995) in this context. It consists of five open questions that students must answer at the end of each week.

The Modified Rosenberg Self-Esteem Scale is a ten item instrument designed to measure self-esteem or self-motivation that was adapted for computer programming by Bergin and Reilly (2006). In its original version it was appropriate to measure self-esteem in any particular situation in diverse areas. In the modified version it was developed to measure self-esteem in working with computer programming. In the study, it was used to confirm the efficiency of the proposed approach by evaluating students' motivation toward the programming process at the end of the course. Bergin and Reilly (2006), Bergin and Reilly (2005), and Kranch (2010) described its use in similar contexts. Bergin and Reilly (2006) used it with 123 freshmen students of introductory programming courses from four institutions in the Republic of Ireland, and validated it with a Cronbach's alpha value of .91.

Schmitt and Allik (2005) argued about the instrument considering it psychometrically valid because of its demonstration of maintaining reliability and factor structure in a study worked in 53 nations. In the study the Cronbach alpha's mean among all nations was of .81 going from .45 in Congo to .90 in Israel and United Kingdom. Internal reliability among all nations was also verified "with an overall Guttman split-half reliability of .73" (p. 629). Other validity measures were also addressed in the referred study. Permission of use for this instrument is granted by the University of Maryland's Department of Sociology through the following address: http://www.bsos.umd.edu/socy/Research/rosenberg.htm.

The Ramalingam-Wiedenbeck Computer Self-Efficacy Scale is a 25 item instrument adapted for the study using the Java programming version developed by Askar and Davenport (2009). It was originally created with 32 items for C++ programming according to Ramalingam and Wiedenbeck (1998). It is to assess the computer self-efficacy of participants.

The Java version is the one used for the study but adapted by deleting all the not applicable items. The instrument was validated in its original version as described by Ramalingam and Wiedenbeck (1998) with 421 students of an introductory C++ programming course. Its overall reliability was calculated with a Cronbach's alpha of .98. In the Java programming version Askar and Davenport (2009) demonstrated the instrument's validity and reliability with 326 students with a Cronbach's alpha score of .99.

Ethical Considerations

Population and Sample

According to the principles addressed in the Common Rule (Code of Federal Regulations, Chapter 45, part 46, 2009) and the Belmont Report (1979) any study who needs to be worked with human beings must follow the principles of justice (equity), beneficence (risk and benefit analysis), and respect for persons (confidentiality and privacy).

Cluster sampling was the technique used according to the circumstances of the study. This technique has implications associated with the principle of justice because it is impossible to provide equal opportunity of selection to all the freshmen students in computing that take a course in computer programming. For the purposes of the study there were not major concerns with this. All participants had the same opportunity to learn the knowledge and skills associated with the course. No one was affected or had any risk related to the situation exposed. Inclusively, both the students participating in the study and all of the others not participating had the same opportunity to learn the same things. The difference in the study was in the approach used in teaching the class but the final result was the same to all of them in terms of the course requirements and objectives.

The situation described is also associated with the principle of beneficence. According to this situation it is established that no risks exist for the students in their participation in the study, or it is minimum. Students in the experimental group were benefited from the approach used in obtaining the adequate knowledge and skills in a more active way providing for a richer learning experience that speeded up the learning process and that arguably helped them to have a better understanding. However, this is not totally different to have any professor of these courses implementing new techniques or approaches to teach his or her classes, as constantly occur in academic institutions. If minimal risk existed for students having problems with the approach used they had the option to receive additional support in office hours and in more complex situations the teacher had options to make any necessary adjustment. This was not necessary in the course worked with the proposed approach for the study.

The principle of respect for persons was totally guaranteed. Students received information about the study and signed the Consent Form for the study. Other aspects of the study were unknown by them. Each group was totally in a normal class setup. This was true to all the sections of the course. All students had the same normal protection of their privacy, and their confidentiality was addressed under the same circumstances. Their autonomy in determining the best options for them about the study was reflected through their acceptance in the Consent Form. On the other hand, institutional regulations and policies of the study's institution establish clearly the protection of those individuals with diminished autonomy.

In another important aspect, as researchers we have the responsibility to perform ethical research. It is our responsibility to adhere to the ethical guidelines existing for researchers. Our work will be evaluated according to performance and how we addressed the different ethical issues we can find during the investigation (Swanson and Holton, 2005). There are situations in which potential risks to participants in a study can be present if we, as researchers, are not aware of some circumstances.

One important situation of potential risks is associated with the researcher's decisions to work the study in the same organization in which he or she works. In doing so important biases can occur if employees or students in this case decide not to participate in the study. First of all, potential participants for a study must have a clear option to deny their participation. Second, they must have the guarantee to be safe from any decision they make or any situation that can occur that may impede their participation. It is important to avoid being in any situation of conflicts of interest "in which their interests or relationships could reasonably be expected to: (1) impair their objectivity, competence, or effectiveness; or (2) expose the persons or organizations with whom the relationships exist to harm or exploitation" (Academy of Management, 2008).

If they opt to be part of the study their participation must be managed with guarantees of anonymity and confidentiality, as already mentioned. No other people may have access to participants' information or executions through the study. The same occurs with the information of the organization as stated by Cooper and Schindler (2011). They stated that "researchers and assistants protect the confidentiality of the sponsor's information and the anonymity of the participants" (p. 44). If we can establish adequate guarantees to sponsors and participants we are overweighting the risks associated with the study but stronger actions must be part of this. Benefits need to be evidently present as mentioned before. All of these considerations were totally covered in working the study.

A particular situation of the described study was the researcher's direct participation in the study as the facilitator of the proposed approach. Since this can be considered inappropriate for most studies, in this case it was considered totally beneficial in favor of avoiding any bias. The researcher was the developer of the proposed approach. No other professor knew about it throughout the study. The other faculty members that teach the first programming course have used the traditional approach extensively for years. The arrangement of having one of them working with the control group using the traditional approach and the researcher, the developer of the proposed approach, working with the experimental group was optimal in the context of the study. This way any bias of trying to work the proposed approach in the traditional manner by one of the other professors was fully avoided. It was precisely an important aspect of the study to make all the possible arrangements with the proposed approach to improve students' knowledge and skills using unconventional techniques defined as part of the design, especially with the ideas adopted from extreme programming. This implied no risks or bias of having contamination of the treatment for the experimental group because of any concern anyone can had against the techniques used with the proposed approach. In fact for most studies in the literature the researchers are direct participants as implementers of the treatment under investigation. Some examples include White (2006), Harper (2006), Pokorny (2009), and Daly (2011).

Role of the Researcher

For the study it was understood that the experience of the researcher was appropriate. There are always situations that were helpful in improving the general knowledge and skills associated with the research process and, in this case, with the teaching process of computer programming. However, as stated, enough experience had been developed through years working in the academia both as professor and as graduate student.

The study was conducted in a four year university institution. This is the working institution of the researcher, an issue considered an important concern in most studies' situation. The characteristics described of the study make of this situation one of less importance within its context. Another situation justified within the context of the study was the researcher's direct participation as the professor in front of the experimental group, working with the implementation of the proposed approach. The experience developed through years working with freshmen students in courses like the one implemented provided all the needed knowledge and skills to do an optimal work.

There were not issues associated with any possible conflict of interests of the researcher in working with the study. The total benefit of the proposed approach acted in favor of students without any potential risks for them. As a regular faculty member of the department and the institution the researcher worked in the normal situation that occurs in all semesters, with classes commonly assigned as part of the regular schedule and within the basic credits load for a professor.

Data Collection

In the data collection procedures established for the study, the Belmont principle of respect for person expressed in terms of protecting the privacy rights of participants and their confidentiality was followed throughout all stages of the study. The situation of needing to work in the study with real freshmen students in a real setup of the first computer programming course presented the most important guarantee that establish accomplishment of this principle. This was because the policies and regulations of the institution, as occur with any competent and accredited university, establishes clearly the full compliance of the principle as part of the common procedures applicable to all university community members. In this context the procedures used in data collection kept the full accomplishment of the principle because the main activities occurred as part of the normal procedures for the course. Actually there was only one activity that occurred in some way out of the context of the course and it was the focus group. However, this activity was still covered by the general policies and regulations established by the institution. The institution of the study is accredited by the Middle States Commission for Higher Education and the state local General Education Council. Based on the protection procedures existing in it, there were not any situations that revealed any personal situation of students, including their participation in the study and any access to their personal information or performance.

The beneficence principle was also totally followed in working with data collection from participants. As part of the standard procedures established in working with students in classes, students received constantly, for any activity, specific instructions about it. They had at all times the opportunity to clarify any doubt, which can be reasonably cleared, before, throughout or after any activity. All the activities used in the study were considered part of the classes, and they occurred in any other sections of the same classes and in any other classes that were not part of the study. This implied that there were not any physical or psychological risk, anxiety or harm that occurred more than the normal stress or anxiety that caused any normal activity that is commonly part of a course, like been in conference, collaborative or examination activities.

Because the participation of students in the study occurred in the context of the common procedures established as part of the teaching of a normal course, the principle of justice and equity was also fully accomplished. There were not more incentives to students than the final grade they received after course ending as occurred in any course. All students had the same opportunity to achieve the goal of passing the class in the same circumstances. All students had also the same opportunity of meeting with the professor in office hours, or of establishing any contact through E-Mail or chat, in order to receive support with doubt clarifications or extra explanation of any misunderstood concept or procedure. The availability to assist students through office hours and other mechanisms is an obligation of any faculty member teaching in the study's institution as part of their job responsibilities. Also there was not any situation that required from students anything that may cause them to be treated in a different way than others in terms of their participation, or that may cause any difference or special considerations in the collection of data from them.

CHAPTER 4. DATA ANALYSIS AND RESULTS

Introduction

As established the purpose of this study was to examine if the implementation of a new approach for teaching computer programming to freshmen students in computer sciences causes an improvement in the acquisition of programming knowledge and skills, in comparison to students that receive the classes using the established traditional approach. The study designed was performed according to the methodology defined. Two groups were analyzed throughout the first semester of academic year 2012-2013 in the university of the study. The control group had been taught using the traditional approach of teaching programming and the experimental group had been taught with the proposed approach.

Description of the Population and Sample

The Population

The population of the study consists of all the freshmen students taking an introductory course in computer programming in any program related to computing or any other program that requires introductory courses in computer programming. These programs are typically of computer sciences and information systems, among others.

The Sample Frame and Sample

The sample frame was all the freshmen students that are admitted to the Computer Science Program in the university where the study was conducted. The freshmen students admitted to the program are normally 90; however, in August 2012 (the date when the study began) 87 students were admitted to the Computer Science Program.

The final list of students admitted was considered the final sample frame authorized by institution's administration in permitting working the study. All students were high school graduates with ages ranging from 17 to 19. They are normally good students that are admitted from the best in the country according to the admissions procedures established. In Puerto Rico the better students compete for the programs in the only state university in the country consisting of eleven campuses.

The sample groups were originally intended to have up to 20 participants in each. This is directly associated with the maximum number of students that are allowed in each course section for COTI 3101 – Algorithms and Program Development I. This course is the first one in programming offered as part of a four year program in computer sciences. In the study the number of students participating was finally less in each group.

Sampling Procedure, Size and Power

As described, the original intention of the study was to have 20 freshmen students in each group. Their selection was expected to occur automatically random. Actually this was the situation. The selection of participants occurred automatically random because all students determined which course section they preferred to enroll without any intervention from professors or any official of the institution. The situation fits in the description of cluster sampling because the groups or clusters occurred naturally within the context of the population, as described by Plano Clark and Creswell (2008). From a pool of six sections, all students decided the group they wanted to enroll. Then, within each group, they were randomly distributed.

The sections previously selected for the study were finally composed of 19 students in each. However, after the consenting process the structure of the two samples consisted of 18 for the experimental group and 12 for the control group. Throughout the semester some students dropped from the classes or simply disappeared. This situation resulted in 13 participants in the experimental group and ten in the control group. The situation described is included in Table 5.

Table 6. Final Samples Situation

	Experimental Group	Control Group
Enrolled students	19	19
Signed Consent Form	18	12
Dropped from class or disappeared	5	6
Dropped or disappeared who signed	5	2
Completed the final surveys	13	10
Completed posttest who signed	13	10

As described in Chapter 3 both groups were previously assigned to the professors participating in the study as part of the normal procedures established. They did not know the students enrolled in their sections until the first day of classes. The sections assigned to the participating faculty members were from those reserved only for newly admitted students.

As can be seen the number of participants for each groups were small and different form each other. However, this situation did not have important implications in terms of power because the samples were generally homogeneous in terms of the selection procedure used. The procedure used was exactly the intended for the study. This was also the case for the focus group exercise in which the random selection of participants was done as established (homogeneous sampling). However, only eight of the ten students selected for participation in the focus group finally participated.

Summary of Results

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 1.1. What is the difference between students working with the proposed approach in terms of performance as compared to the performance of the students that received instruction using the traditional approach? (Quantitative Approach)

The hypotheses formulated to answer this question were:

$H_0 1.1$: The means of the scores for the two groups are not significantly different.

$H_A 1.1$: The means of the scores for the two groups are significantly different.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of performance.

ResQ 1.2. What is the difference between both groups on the motivation of students toward programming? (Quantitative Approach)

The hypotheses formulated to answer this question were:

$H_0 1.2$: Students using the proposed methodology were similarly motivated or less motivated toward using programming in comparison to those using the traditional approach.

$H_A 1.2$: Students using the proposed methodology were more motivated toward using programming in comparison to those using the traditional approach.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of the student's motivation toward programming.

ResQ 1.3. What is the difference between both groups in the predisposition of students toward continuing in the program? (Quantitative Approach)

The hypotheses formulated to answer this question were:

$H_0 1.3$: Students using the proposed approach were similarly predisposed or less predisposed to continue in the program in comparison to those using the traditional approach.

$H_A 1.3$: Students using the proposed approach were more predisposed to continue in the program in comparison to those using the traditional approach.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of student's predisposition to continue in the program.

ResQ 1.4. How do the students feel in working with the proposed approach? (Qualitative Approach)

Results: Positive expressions were generally made about the experience by students. They were common throughout all the focus group allowing establishing that they felt mostly comfortable with the approach used. In specific expressions one participant referred to situations in which he or she felt less comfortable or felt uncomfortable.

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 2.1. What is the improvement, if any, in the average time needed by students to get confident in programming using the main building blocks and structures required to be learned in the first programming class? (Quantitative Approach)

The hypotheses related to this question were:

> *$H_0$2.1:* The learning of programming concepts for the students using the proposed approach will be similar to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.
>
> *H_A2.1:* The learning of programming concepts for the students using the proposed approach will be faster compared to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

ResQ 2.2. Based on the Critical Incident Questionnaire (CIQ) instrument for formative assessment, how much do the computer science freshmen students participating in the study using the proposed approach to teaching programming benefit constantly from using it? (Qualitative Approach)

$H_0$2.2: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly similar or negative about their understanding of the concepts presented.

H_A2.2: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly positive about their understanding of the concepts presented.

Results: The null hypothesis must be rejected. The majority of the expressions done by participants in the experimental group through the CIQ were positive in those weeks in which aspects of the proposed approach were integrated.

ResQ 2.3. How much time is needed to get confident with the different building blocks and structures while the understanding acquired according to the CIQ is improved using the proposed approach? (Mixed Approach)

$H_0$2.3: The time needed to get confident with the different building blocks and structures will appear to be similar or higher and the understanding acquired as reflected in the CIQ will appear to be similar or lower while students learn with the proposed approach.

H_A2.3: The time needed to get confident with the different building blocks and structures will appear to be reduced and the understanding acquired as reflected in the CIQ will appear to be higher while students learn with the proposed approach.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of the time needed to get confident with the different building blocks and structures. It can be argued that an understanding of the building blocks and structures was occurred according to the expressions made through the CIQ but it had not apparently any effect in shortening the general time needed to learn them.

ResQ 2.4. How much significant is the difference between the two groups of students in performance and how much positive is it in favor of the proposed approach? (Quantitative Approach)

The following are the hypotheses related to this question. As can be seen they are about the students' performance by mean of the final score obtained in the classes comparing the control and the experimental groups.

> *$H_0$2.4:* The means of the test scores for the two groups are not significantly different or the mean for the group working with the proposed approach is significantly lower.
>
> *H_A2.4:* The mean test scores for the group working with the proposed approach is significantly higher.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of performance.

ResQ 2.5. How much significant is the difference between the two groups of students in motivation toward programming and how much positive is it in favor of the proposed approach? (Quantitative Approach)

The hypotheses related to this question are the following.

> *$H_0$1.2:* Students using the proposed methodology were similarly motivated or less motivated toward using programming in comparison to those using the traditional approach.

$H_A1.2$: Students using the proposed methodology were more motivated toward using programming in comparison to those using the traditional approach.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of the student's motivation toward programming.

ResQ 2.6. How much significant is the difference between the two groups of students in the predisposition toward continuing in the program and how much positive is it in favor of the proposed approach? (Quantitative Approach)

The hypotheses related to this question are the following.

$H_01.3$: Students using the proposed approach were similarly predisposed or less predisposed to continue in the program in comparison to those using the traditional approach.

$H_A1.3$: Students using the proposed approach were more predisposed to continue in the program in comparison to those using the traditional approach.

Results: The null hypothesis must be retained. There was no significant difference between both groups in terms of student's predisposition to continue in the program.

Details of Analysis and Results

Statistical Assumptions Recognized

The statistical methodologies finally implemented in comparing the results from the two groups were based in using non parametric tests. Non parametric tests were used because some of the assumptions required for the use of parametric tests were not met.

- The assumption of working with independent samples was met. The two groups used for the study were totally independent and randomly selected.

- The assumption of having normally distributed data was not met in both groups. The histograms for the two groups presented in Figure 3 for the control group and in Figure 4 for the experimental group show the lack of normality in the behavior of the data obtained.
- The assumption of having equal variances was also not met. The variance for the control group was 113.163 and for the experimental group it was 123.295.

The non parametric test used in most of the situations was the Mann-Whitney U test. This test is appropriate for comparison of two independent groups. It is good for working with both interval and ordinal data, and it is comparable with the *t*-Test. Also it is appropriate when using small samples (less than 20 subjects).

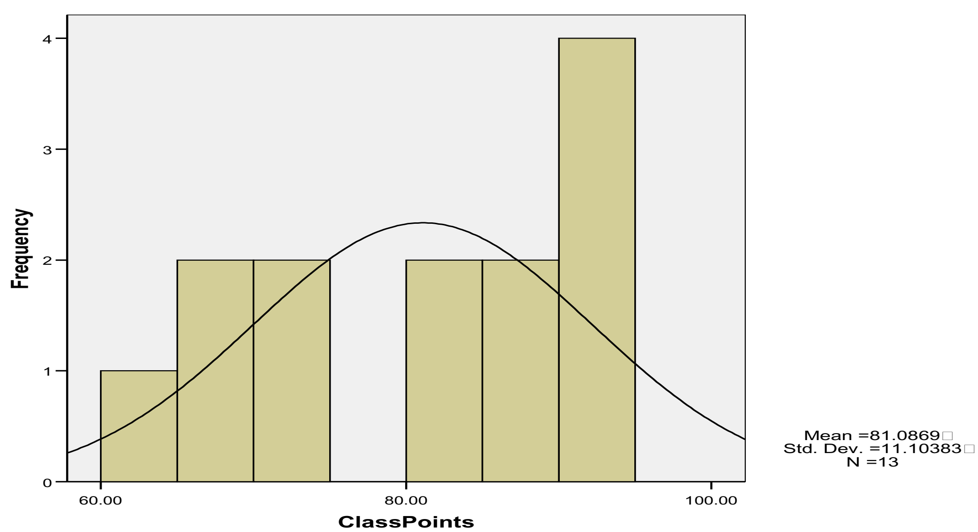

Figure 3. Histogram for the control group

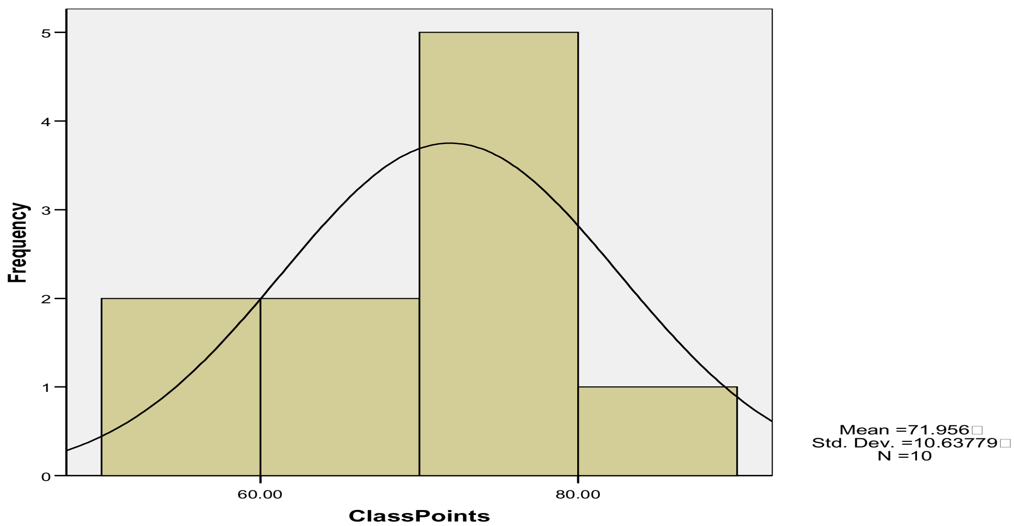

Figure 4. Histogram for the experimental group

Research Questions

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 1.1. What is the difference between students working with the proposed approach in terms of performance as compared to the performance of the students that received instruction using the traditional approach? (Quantitative Approach)

The hypotheses formulated to answer this question were:

H_0*1.1:* The means of the scores for the two groups are not significantly different.

H_A*1.1:* The means of the scores for the two groups are significantly different.

In analyzing the data obtained from the students' scores after classes ended the non parametric Mann-Whitney U test was used. Table 6 shows the data obtained from the two groups. Table 7 and Table 8 show the results of running the referred test using SPSS. Figure 5 is a line chart that shows a graphical representation of the behavior demonstrated by the data.

The results obtained from the Mann-Whitney U test indicate that the null hypothesis must be retained because based on a significant level of $p < 0.05$ does not exist a statistically significant difference between the two groups.

Table 7. Class Final Scores Obtained After End of Semester for both the Control and the Experimental Groups

Experimental	Control
85	35
75	70
80	52
48	54
43	57
74	31
43	83
97	43
78	48
66	58
55	
91	
64	

Table 8. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Class Scores

Ranks

	Group	N	Mean Rank	Sum of Ranks
FinalScore	Experimental	13	14.42	187.50
	Control	10	8.85	88.50
	Total	23		

Table 9. Second Part of the Results Obtained After Running the Mann-Whitney U Test for the Class Scores

Test Statistics(b)

	FnalScore
Mann-Whitney U	33.500
Wilcoxon W	88.500
Z	-1.956
Asymp. Sig. (2-tailed)	.050
Exact Sig. [2*(1-tailed Sig.)]	.049(a)

a Not corrected for ties.
b Grouping Variable: Group

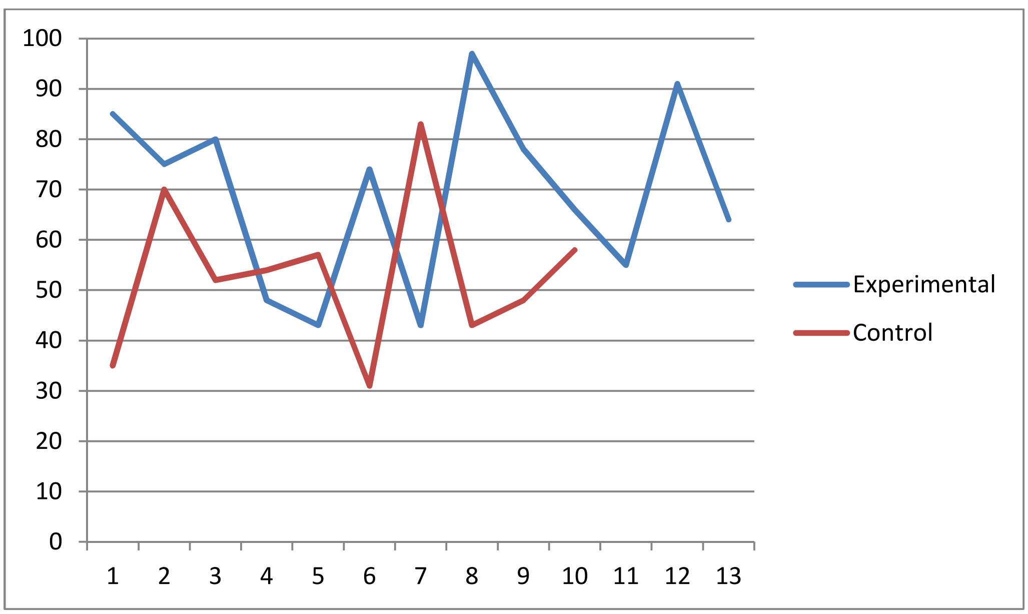

Figure 5. Line chart presenting graphically the behavior of the final scores for both groups

ResQ 1.2. What is the difference between both groups on the motivation of students toward programming? (Quantitative Approach)

The hypotheses formulated to answer this question were:

$H_0$1.2: Students using the proposed methodology were similarly motivated or less motivated toward using programming in comparison to those using the traditional approach.

H_A1.2: Students using the proposed methodology were more motivated toward using programming in comparison to those using the traditional approach.

Table 10. Response Data Obtained from the Modified Rosenberg Self-Esteem Scale Questionnaire Administered to the Experimental Group

	SE1	SE2	SE3	SE4	SE5	SE6	SE7	SE8	SE9	SE10
S01	4	3	3	4	3	4	4	3	4	4
S02	3	1	3	3	2	3	3	2	3	3
S03	4	3	4	4	4	4	4	3	4	4
S04	4	4	4	3	3	4	4	4	4	4
S05	4	3	3	3	4	4	4	4	4	4
S06	3	2	2	3	3	2	2	2	3	3
S07	4	3	3	3	4	4	3	3	4	4
S08	4	4	4	4	4	4	4	3	4	4
S09	4	2	3	2	2	1	4	2	4	4
S10	3	3	3	3	3	3	3	2	3	4
S11	4	3	3	4	3	4	4	3	4	4
S12	4	2	3	2	4	3	3	3	3	3
S13	4	4	4	4	4	4	4	4	4	4

In analyzing the data obtained from the Modified Rosenberg Self-Esteem Scale after classes ended the non parametric Mann-Whitney U test was used. Table 9 shows the data obtained from the experimental group and Table 10 shows the data from the control group. Table 11 and Table 12 show the results of running the referred test using SPSS.

The results obtained from the Mann-Whitney U test run using the data obtained from the responses to the Modified Rosenberg Self-Esteem Scale questionnaire indicate that for 9 of the 10 responses the null hypothesis must be retained. This is based on a significant level of p = 0.05. In this case a statistically significant difference between the two groups was not found.

Table 11. Response Data Obtained from the Modified Rosenberg Self-Esteem Scale Questionnaire Administered to the Control Group

	SE1	SE2	SE3	SE4	SE5	SE6	SE7	SE8	SE9	SE10
S01	3	1	2	1	1	1	1	1	3	4
S02	4	3	3	4	4	4	4	3	4	4
S03	3	3	3	3	3	3	3	3	3	3
S04	3	2	3	3	3	3	4	2	3	3
S05	3	2	3	4	4	4	3	3	4	3
S06	4	3	3	3	4	4	3	3	4	4
S07	2	1	3	4	1	4	4	2	1	1
S08	2	2	2	2	1	2	3	1	3	3
S09	3	3	3	3	3	4	3	3	4	4
S10	3	3	3	2	4	3	4	3	4	4

Table 12. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Modified Rosenberg Self-Esteem Scale Questionnaire

Ranks

	Group	N	Mean Rank	Sum of Ranks
VAR00001	Experimental	13	15.08	196.00
	Control	10	8.00	80.00
	Total	23		
VAR00002	Experimental	13	13.65	177.50
	Control	10	9.85	98.50
	Total	23		
VAR00003	Experimental	13	13.85	180.00
	Control	10	9.60	96.00
	Total	23		
VAR00004	Experimental	13	12.88	167.50
	Control	10	10.85	108.50
	Total	23		
VAR00005	Experimental	13	12.88	167.50
	Control	10	10.85	108.50
	Total	23		
VAR00006	Experimental	13	12.58	163.50
	Control	10	11.25	112.50
	Total	23		
VAR00007	Experimental	13	13.08	170.00
	Control	10	10.60	106.00
	Total	23		

Table 13. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Modified Rosenberg Self-Esteem Scale Questionnaire (Continued)

Ranks

	Group	N	Mean Rank	Sum of Ranks
VAR00008	Experimental	13	13.46	175.00
	Control	10	10.10	101.00
	Total	23		
VAR00009	Experimental	13	13.12	170.50
	Control	10	10.55	105.50
	Total	23		
VAR00010	Experimental	13	13.46	175.00
	Control	10	10.10	101.00
	Total	23		

ResQ 1.3. What is the difference between both groups in the predisposition of students toward continuing in the program? (Quantitative Approach)

The hypotheses formulated to answer this question were:

$H_0 1.3$*:* Students using the proposed approach were similarly predisposed or less predisposed to continue in the program in comparison to those using the traditional approach.

$H_A 1.3$*:* Students using the proposed approach were more predisposed to continue in the program in comparison to those using the traditional approach.

Table 14. Second Part of the Results Obtained After Running the Mann-Whitney U Test for the Modified Rosenberg Self-Esteem Scale Questionnaire

Test Statistics(b)

	VAR00001	VAR00002	VAR00003	VAR00004	VAR00005	VAR00006	VAR00007	VAR00008	VAR00009	VAR00010
Mann-Whitney U	25.000	43.500	41.000	53.500	53.500	57.500	51.000	46.000	50.500	46.000
Wilcoxon W	80.000	98.500	96.000	108.500	108.500	112.500	106.000	101.000	105.500	101.000
Z	-2.775	-1.429	-1.836	-.764	-.763	-.519	-.971	-1.286	-1.050	-1.413
Asymp. Sig. (2-tailed)	.006	.153	.066	.445	.446	.603	.332	.198	.294	.158
Exact Sig. [2*(1-tailed Sig.)]	.012(a)	.186(a)	.148(a)	.483(a)	.483(a)	.648(a)	.410(a)	.257(a)	.376(a)	.257(a)

a Not corrected for ties.
b Grouping Variable: Group

The Ramalingam-Wiedenbeck Computer Self-Efficacy Scale questionnaire was responded by students from both groups in the last class of the semester. The data obtained from the responses provided were analyzed using the non parametric Mann-Whitney U test. The experimental group data obtained from this questionnaire is presented in Table 13 and the control group data is presented in Table 14. The reliability of using the tool with the experimental group was established with a Cronbach's Alpha of .96 while with the control group it was established in .95. The results obtained after running the Mann-Whitney U test using SPSS with the data from the questionnaire are presented in Table 15 and Table 16.

In 23 of the 25 items analyzed using the Mann-Whitney U test for comparison of the responses from both groups to the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale questionnaire, the test indicated that the null hypothesis must be retained based on a significant level p value of 0.05. A statistically significant difference between the two groups was not found.

ResQ 1.4. How do the students feel in working with the proposed approach? (Qualitative Approach)

After the classes ended and the final exam was completed by students, a focus group was cited. It was worked some days after the exam ensuring the professor enough time to have the final class grades available. The focus group ran for one hour and 26 minutes. It was made in Spanish, the native language of all students. In general, students made interesting expressions about their experience in the class. It is important to establish that some expressions were translated from Spanish to English.

Table 15. Response Data Obtained from the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire Administered to the Experimental Group

	PS E1	PS E2	PS E3	PS E4	PS E5	PS E6	PS E7	PS E8	PS E9	PS E10	PS E11	PS E12	PS E13	PS E14	PS E15	PS E16	PS E17	PS E18	PS E19	PS E20	PS E21	PS E22	PS E23	PS E24	PS E25
S01	6	5	5	7	7	7	7	6	6	5	4	5	6	7	4	7	7	7	4	5	5	6	7	7	5
S02	5	5	5	7	4	4	7	6	4	4	4	3	5	6	4	5	7	4	4	4	3	2	3	2	5
S03	6	7	5	4	7	6	7	5	4	4	5	6	7	7	6	6	7	7	6	4	3	6	3	7	7
S04	6	6	6	7	7	7	6	5	5	6	6	7	7	6	6	7	7	7	6	6	6	7	5	7	6
S05	6	7	7	7	7	7	7	6	7	6	7	7	7	7	7	7	7	6	4	5	5	7	7	7	7
S06	4	4	4	5	5	5	5	4	4	3	3	4	4	4	3	3	4	5	4	4	4	4	3	3	3
S07	6	5	5	7	7	7	7	6	6	5	5	6	6	7	6	7	5	7	6	6	7	5	7	6	7
S08	7	7	6	7	7	7	7	7	5	7	7	7	6	7	7	7	7	7	7	7	7	5	7	7	7
S09	5	5	5	7	7	7	6	6	5	4	5	5	6	6	6	4	4	7	4	6	4	6	6	7	6
S10	5	5	5	7	7	7	1	1	3	4	4	4	5	6	3	4	4	6	4	4	3	4	5	5	4
S11	6	6	5	7	6	6	4	6	7	6	5	6	7	7	5	7	7	7	5	7	5	6	6	7	5
S12	7	7	7	6	5	5	6	4	2	2	3	2	5	6	3	6	5	7	4	3	2	4	2	4	1
S13	7	7	7	7	7	7	7	7	6	7	7	6	7	7	7	7	7	7	7	7	7	6	7	7	7

Table 16. Response Data Obtained from the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire Administered to the Control Group

	PS E1	PS E2	PS E3	PS E4	PS E5	PS E6	PS E7	PS E8	PS E9	PS E10	PS E11	PS E12	PS E13	PS E14	PS E15	PS E16	PS E17	PS E18	PS E19	PS E20	PS E21	PS E22	PS E23	PS E24	PS E25
S01	3	3	4	7	6	2	2	5	4	4	1	2	2	1	7	6	4	6	4	2	1	1	1	1	1
S02	5	6	5	7	7	6	7	6	6	5	5	6	5	6	4	5	6	7	5	5	4	4	5	5	6
S03	4	4	4	7	5	5	6	5	4	4	3	4	4	5	4	5	5	5	4	5	4	3	3	4	4
S04	4	5	5	7	6	5	5	5	4	5	4	4	5	5	5	6	5	6	5	5	4	4	5	6	6
S05	5	6	5	7	7	6	6	6	6	5	5	7	6	6	4	5	7	7	5	5	5	6	4	4	6
S06	6	6	6	7	7	7	7	7	4	7	4	5	6	5	7	7	7	7	5	6	2	7	4	6	6
S07	6	5	4	7	7	7	7	7	7	4	7	1	1	7	7	5	4	7	4	3	1	5	1	7	4
S08	5	4	4	7	5	3	4	3	3	4	5	4	3	4	5	4	3	5	4	4	1	1	1	2	3
S09	5	5	5	7	7	6	6	6	4	5	5	5	3	4	4	5	6	6	6	4	4	7	6	7	5
S10	7	7	7	7	7	7	7	4	5	7	6	6	5	5	4	7	7	7	7	6	7	5	6	5	5

Table 17. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire

Ranks

	Group	N	Mean Rank	Sum of Ranks
VAR00001	Experimental	13	14.19	184.50
	Control	10	9.15	91.50
	Total	23		
VAR00002	Experimental	13	13.69	178.00
	Control	10	9.80	98.00
	Total	23		
VAR00003	Experimental	13	13.88	180.50
	Control	10	9.55	95.50
	Total	23		
VAR00004	Experimental	13	10.85	141.00
	Control	10	13.50	135.00
	Total	23		
VAR00005	Experimental	13	12.23	159.00
	Control	10	11.70	117.00
	Total	23		
VAR00006	Experimental	13	13.69	178.00
	Control	10	9.80	98.00
	Total	23		
VAR00007	Experimental	13	12.65	164.50
	Control	10	11.15	111.50
	Total	23		

Table 18. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire (Continued)

Ranks

	Group	N	Mean Rank	Sum of Ranks
VAR00008	Experimental	13	12.15	158.00
	Control	10	11.80	118.00
	Total	23		
VAR00009	Experimental	13	12.65	164.50
	Control	10	11.15	111.50
	Total	23		
VAR00010	Experimental	13	11.77	153.00
	Control	10	12.30	123.00
	Total	23		
VAR00011	Experimental	13	12.62	164.00
	Control	10	11.20	112.00
	Total	23		
VAR00012	Experimental	13	13.38	174.00
	Control	10	10.20	102.00
	Total	23		
VAR00013	Experimental	13	15.46	201.00
	Control	10	7.50	75.00
	Total	23		
VAR00014	Experimental	13	15.35	199.50
	Control	10	7.65	76.50
	Total	23		

Table 19. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire (Continued)

Ranks

	Group	N	Mean Rank	Sum of Ranks
VAR00015	Experimental	13	11.96	155.50
	Control	10	12.05	120.50
	Total	23		
VAR00016	Experimental	13	13.27	172.50
	Control	10	10.35	103.50
	Total	23		
VAR00017	Experimental	13	13.31	173.00
	Control	10	10.30	103.00
	Total	23		
VAR00018	Experimental	13	12.81	166.50
	Control	10	10.95	109.50
	Total	23		
VAR00019	Experimental	13	11.96	155.50
	Control	10	12.05	120.50
	Total	23		
VAR00020	Experimental	13	13.35	173.50
	Control	10	10.25	102.50
	Total	23		
VAR00021	Experimental	13	13.96	181.50
	Control	10	9.45	94.50
	Total	23		

Table 20. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire (Continued)

Ranks

	Group	N	Mean Rank	Sum of Ranks
VAR00022	Experimental	13	13.27	172.50
	Control	10	10.35	103.50
	Total	23		
VAR00023	Experimental	13	14.35	186.50
	Control	10	8.95	89.50
	Total	23		
VAR00024	Experimental	13	13.96	181.50
	Control	10	9.45	94.50
	Total	23		
VAR00025	Experimental	13	13.62	177.00
	Control	10	9.90	99.00
	Total	23		

Specific expressions associated with this research question were made by some students in the focus group. They mostly emerged from the answers to two of the questions made. These were:

- What do you think about your learning experience in the class?
- How comfortable were you during the course with the approach used?

Table 21. Second Part of the Results Obtained After Running the Mann-Whitney U Test for the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire

Test Statistics(b)

	VAR00001	VAR00002	VAR00003	VAR00004	VAR00005	VAR00006	VAR00007	VAR00008	VAR00009	VAR00010
Mann-Whitney U	36.500	43.000	40.500	50.000	62.000	43.000	56.500	63.000	56.500	62.000
Wilcoxon W	91.500	98.000	95.500	141.000	117.000	98.000	111.500	118.000	111.500	153.000
Z	-1.839	-1.415	-1.625	-1.589	-.220	-1.457	-.564	-.129	-.543	-.192
Asymp. Sig. (2-tailed)	.066	.157	.104	.112	.826	.145	.573	.897	.587	.847
Exact Sig. [2*(1-tailed Sig.)]	.077(a)	.186(a)	.131(a)	.376(a)	.879(a)	.186(a)	.605(a)	.927(a)	.605(a)	.879(a)

	VAR00011	VAR00012	VAR00013	VAR00014	VAR00015	VAR00016	VAR00017	VAR00018	VAR00019	VAR00020
Mann-Whitney U	57.000	47.000	20.000	21.500	64.500	48.500	48.000	54.500	64.500	47.500
Wilcoxon W	112.000	102.000	75.000	76.500	155.500	103.500	103.000	109.500	155.500	102.500
Z	-.511	-1.138	-2.856	-2.807	-.032	-1.069	-1.126	-.745	-.033	-1.112
Asymp. Sig. (2-tailed)	.609	.255	.004	.005	.975	.285	.260	.456	.974	.266
Exact Sig. [2*(1-tailed Sig.)]	.648(a)	.284(a)	.004(a)	.005(a)	.976(a)	.313(a)	.313(a)	.522(a)	.976(a)	.284(a)

Table 22. Second Part of the Results Obtained After Running the Mann-Whitney U Test for the Ramalingam-Wiedenbeck Computer Self-Efficacy Scale Questionnaire (Continued)

	VAR00021	VAR00022	VAR00023	VAR00024	VAR00025
Mann-Whitney U	39.500	48.500	34.500	39.500	44.000
Wilcoxon W	94.500	103.500	89.500	94.500	99.000
Z	-1.607	-1.043	-1.918	-1.656	-1.329
Asymp. Sig. (2-tailed)	.108	.297	.055	.098	.184
Exact Sig. [2*(1-tailed Sig.)]	.115(a)	.313(a)	.057(a)	.115(a)	.208(a)

a Not corrected for ties.
b Grouping Variable: Group

Positive expressions were generally made about the experience by students. One of them expressed the following ideas:

> I truly think that the class was beginner friendly. When the course began, I didn't know much about programming, but after the topics were introduced, I basically understood everything. Initially, everything appeared to me… not easy… I learned that it's not easy to understand. You need to obviously attend classes and pay close attention. The course seemed to me well structured, everything had a logical order, and relatively included everything I expected it to have. This type of structure is something that I cannot find on the Internet. The course was organized, and this is what helped me understand.

To these ideas the only female student of the group reacted:

> I agree with him when he referred to learning about computer programming just by reading or listening to stuff on the Internet. I thought I could find on the Internet ways to learn programming, but this was just a painful experience. I learned that it's much easier to learn computer programming when you are part of a course that includes Power Points presentations that adequate information discussed in class.

A third student established that:

> I think that the class developed the topics in a comfortable and smooth pace. You don't need to learn everything in a rush like in other classes. For example, in the pre calculus you need to learn many concepts and theories in one same day and then empty all the information on an exam. In this course, one needs to simply learn the material gradually. In each class you add to what you know and what you learned the day before. You are always learning and using what you learn. I felt confident in class.

Another one expressed his experience in the following terms:

> I have some previous knowledge of programming from high school classes, but I didn't understand some details that I eventually clarified in this class. For example, some details that I understood in this class were the files and how computers work. An interesting aspect of the class was it motivated us to think beyond. For example, an awesome exercise is the one in which I ran a program and was able to see the total sum and variables. This was something I had never done before. In addition, throughout this course I learned about logic and the diverse things that can be done when using logic. I liked it… I really enjoyed this course.

A relatively negative experience was mentioned by a student in these words:

> The situation is that I share the same opinion expressed by my partners because it is not easy for me to study… but now I understand most of the topics discussed in class. I could not work on many things on my own. I sometimes looked for help. For example, some of the programs I made did not work because I was usually nervous or because I said to myself, "I cannot do this! I will not do this well!" However, when I saw my classmates programs, I realized that I also had those same ideas in my mind. That all I really needed to do was put them into practice. Thus, I can say, I learned, but my frustration got in the way and did not allow me to perform as well. Despite my negative attitude, I must seriously say that I learned anyway. I think that I will always remember the acquired knowledge.

In different expressions throughout the focus group the participants referred to situations in which they felt comfortable and others in which they weren't. Some of them expressed that they believe the intensive exercises were excellent in helping them to understand concepts and to develop programming skills. Other referred to a situation occurred in a specific topic in which the time was so short in order to allow them to get confidence appropriately with the related concepts and skills.

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 2.1. What is the improvement, if any, in the average time needed by students to get confident in programming using the main building blocks and structures required to be learned in the first programming class? (Quantitative Approach)

The hypotheses related to this question were:

> *$H_0$2.1:* The learning of programming concepts for the students using the proposed approach will be similar to the learning of programming concepts for those using the traditional approach according to the average time needed to be confident in programming using the main building blocks and structures required to be learned in the first programming class.
>
> *H_A2.1:* The learning of programming concepts for the students using the proposed approach will be faster compared to the learning of programming concepts for those using the traditional approach according to the average time needed to get confident in programming using the main building blocks and structures required to be learned in the first programming class.

In determining if there was any improvement in the time needed to work with the main areas of discussion in the classes the methodology was implemented with the final result obtained as shown in Table 17. Exclusive topics and topics that were worked simultaneously were already excluded from the table.

In Table 18 the main areas discussed were included with the average weeks in which each of them were discussed. This additional information was provided in order to allow for discussion of additional interesting aspects reflected in the data that may be of importance for the study in trying to explain the phenomena occurred.

The analysis of the data associated with the weeks in which topics were worked was done using independent samples *t*-Test which was appropriate for comparison between groups in this case. Table 19 and Table 20 show the results obtained after running the *t*-Test as generated by SPSS.

In comparing the means of the week amounts associated with the time in which the different topics were worked the t-Test indicated that, based on a significant level of $p=0.05$, there was not a general difference between the two groups. This implies that the null hypothesis must be retained.

Table 23. Main Areas and Topics Covered and the Weeks in Which They Were Discussed for Both Groups under Study

Areas	Topic	Control	Experimental
Analysis and Design	Flowchart	6	4
	Program	4	4
Basics	Variable	5	5
	Declaration	5	5
	Data Type	5	5
	Inicialization	5	5
	Constant	5	6
	String objects	5	8
Output	*System.out.println*	5	5
Input	Scanner class and console input	5	10
Selection Control	Selection structure – *if*	6	10
	Complex *ifs*	7	13
	Nested *ifs*	7	14
Loops	*while* loop	11	9
	Counter controlled loops	11	5
	Sentinel controlled loops	11	9
	Response controlled loops	11	9
	Counter	11	5
	Acumulator	11	5
	do – while	11	14
	for loop	11	6
Methods	Methods	16	17
Files	Files	12	16

Table 24. Main Areas Covered and the Average Weeks in Which They Were Discussed For Both Groups under Study

Areas	Control	Experimental
Analysis and Design	5.00	4.00
Basics	5.00	5.67
Output	5.00	5.00
Input	5.00	10.00
Selection Control	6.67	12.33
Loops	11.00	7.75
Methods	16.00	17.00
Files	12.00	16.00

Table 25. Group Statistics of the Weeks in Which Topics Were Worked

Group Statistics

	Groups	N	Mean	Std. Deviation	Std. Error Mean
Weeks	Experimental	23	8.22	4.067	.848
	Control	23	8.09	3.370	.703

ResQ 2.2. Based on the Critical Incident Questionnaire (CIQ) instrument for formative assessment, how much do the computer science freshmen students participating in the study using the proposed approach to teaching programming benefit constantly from using it? (Qualitative Approach)

$H_0 2.2$: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly similar or negative about their understanding of the concepts presented.

$H_A 2.2$: The expressions that students make through the CIQ about their experience in class, while learning with the proposed approach, will be constantly positive about their understanding of the concepts presented.

Some of the expressions made by the participants in answering the questions of the CIQ are included translated in Tables 21 to 25. Because questions were submitted to students in English but they were allowed to answer them in their language of preference, the expressions were translated from Spanish to English. It is important to recognize that they made informal expressions. This is an expected situation within the context of using the CIQ as an instrument for formative assessment, as explained in Chapter 3.

In order to be more specific in presenting results associated mostly with elements of the implementation of the proposed approach, the following comments (Tables 21, 22, 23, 24 and 25) are from the responses to the CIQ in the weeks in which the most important of those elements were worked. These elements are programming language immersion exercises, programming topic immersion exercises and extreme programming exercises (including pair programming or collaborative programming, and intensive exercises). The integration of these activities occurred mostly in the first half of the semester. That is because there are no results presented from the CIQ expressions from the second half of the semester.

Table 26. Independent Samples *T*-Test of the Weeks in Which Topics Were Worked

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Weeks	Equal variances assumed	.382	.540	.118	44	.906	.130	1.101	-2.089	2.350
	Equal variances not assumed			.118	42.530	.906	.130	1.101	-2.091	2.352

Table 27. Some CIQ Expressions – September 12, 2013

Elements used	Question	Expressions
Programming language immersion Programming topic immersion	"At what moment in the class this week did you feel most engaged with what was happening?" (Brookfield, 2006)	At all times… Ever… When the professor starts a new topic involving key skills in programming… I felt most engaged as a leaner when we started to program in class…
	"At what moment in the class this week did you feel most distanced from what was happening?" (Brookfield, 2006)	Never… At first, because I dint know the material… When I received a program for the first time… During the first programming exercise…
	"What action that anyone (teacher or student) took in class this week did you find most affirming and helpful?" (Brookfield, 2006)	That of my classmates since we can all share our knowledge… When my classmates started to ask questions that helped me understand better…
	"What action that anyone (teacher or student) took in class this week did you find most puzzling or confusing?" (Brookfield, 2006)	I don't know… When the professor first started, but then he cleared my doubt… When we start programming, it was a little bit confusing and complicated…
	"What about the class this week surprised you the most?" (Brookfield, 2006)	Beginning with a simple programming algorithm… Try to understand how to make a program and its parts… The complication of programming… I was surprised at not wanting to leave at the end of the class. Haha… Running a program and seeing it works successfully…

Table 28. Some CIQ Expressions – September 19, 2013

Elements used	Question	Expressions
Programming language immersion Programming topic immersion Extreme programming (collaborative)	"At what moment in the class this week did you feel most engaged with what was happening?" (Brookfield, 2006)	Almost at all times. I learned interesting things… When we got more into programming and explaining the different meaning of codes and variables, etc… Once I started experimenting at the program changing variables and some other things… When I start writing the programming language. When I feel that I'm putting my knowledge to practice…
	"At what moment in the class this week did you feel most distanced from what was happening?" (Brookfield, 2006)	At not any moment… When the professor showed us new programs and i couldn't figure it out right away… Sometimes I don't understand but later I do with the help of my partners, and I do the job… When I was lost of why my program wasn't giving me the correct answer… In programming…
	"What action that anyone (teacher or student) took in class this week did you find most affirming and helpful?" (Brookfield, 2006)	My classmates helped me with my problems in the program… When someone explained to me calmly… Everybody help me when I'm lost… I have a little group where we try decipher every little error, meaning and process the professor gives us… Working in the groups and communicating the results…
	"What action that anyone (teacher or student) took in class this week did you find most puzzling or confusing?" (Brookfield, 2006)	Nothing up to now. I still understand… When explaining the System.out.printf… None… A classmate said something too advance for the class we are taking… When programming…
	"What about the class this week surprised you the most?" (Brookfield, 2006)	The specific order that the program shows you its results… Begining programming something, although simple and short… That I was able to understand very quickly the material… How tiny little details can change an entire format of a program… That I understand more easily the things… That I did the exercise correctly without problems… The complexity of the theme…

Table 29. Some CIQ Expressions – September 26, 2013

Elements used	Question	Expressions
Programming topic immersion Extreme programming (pairs and collaborative)	"At what moment in the class this week did you feel most engaged with what was happening?" (Brookfield, 2006)	Putting the code to work… This week it was harder to learn but with practice I know I will succeed… In moments were my doubts were cleared up… When we were working in pairs at one point in class…
	"At what moment in the class this week did you feel most distanced from what was happening?" (Brookfield, 2006)	Figuring out how to organize the code… Never… When i did not understand why my program wasnt working… In programming based on a flowchart…
	"What action that anyone (teacher or student) took in class this week did you find most affirming and helpful?" (Brookfield, 2006)	Making code from memory… The professor showing the right code we needed to do… The practice of making programs in class… My closest classmate helped me correct some errors in my answers… One of my classmates helped me with something I didn't heard…
	"What action that anyone (teacher or student) took in class this week did you find most puzzling or confusing?" (Brookfield, 2006)	For me everything… None… Making flowcharts for programs… This week I didn't felt confused by an action of one of my classmates…
	"What about the class this week surprised you the most?" (Brookfield, 2006)	That I couldn't completed the quiz… How difficult it was to solve a problem… One part of the program was right but the second part kept being wrong… The problems at first but after some analysis, I got the hang of it…

Table 30. Some CIQ Expressions – October 3, 2013

Elements used	Question	Expressions
Programming language immersion Programming topic immersion Extreme programming (pairs and collaborative)	"At what moment in the class this week did you feel most engaged with what was happening?" (Brookfield, 2006)	Entire class… From beginning to end… When I had my doubts about flowcharts and tried to find a logical way for me to organize it… When interacting with the teacher… Throughout all the class due to the dynamic explanations… At all times throughout the class today. It was so interesting…
	"At what moment in the class this week did you feel most distanced from what was happening?" (Brookfield, 2006)	Very little time… When I do not understand something… Never… At this moment I feel distanced because I don't understand a lot of things… At not any moment. I was involved at al times… System.out.print("none");…
	"What action that anyone (teacher or student) took in class this week did you find most affirming and helpful?" (Brookfield, 2006)	Classmates help me out… The collaboration… The questions of my classmates… The professor explaining what to do… Discussing the problems and interchanging ideas… The discussion between classmates…
	"What action that anyone (teacher or student) took in class this week did you find most puzzling or confusing?" (Brookfield, 2006)	None… When the professor did not discuss a method… New topics not previously seen… Trying to figure out what was wrong in our code…
	"What about the class this week surprised you the most?" (Brookfield, 2006)	The problems with the *for*… How there are many ways of organizing a program to solve the same problem… I understand more… Everything surprised me… The introduction of new functions… The different ways to solve a problem…

Table 31. Some CIQ Expressions – October 17, 2013

Elements used	Question	Expressions
Programming language immersion Extreme programming (pairs and intensive exercises)	"At what moment in the class this week did you feel most engaged with what was happening?" (Brookfield, 2006)	When I understood a new programming style… All class… When I understand all the intensive exercises… In all the class because of the intensive exercises working in pairs and the discussion, among other ways of teaching the class… When solving exercises… Once new things were introduced…
	"At what moment in the class this week did you feel most distanced from what was happening?" (Brookfield, 2006)	None… When I was lost due to tiny errors… Never… When I don't understand some parts of the code… I didn't felt distanced…
	"What action that anyone (teacher or student) took in class this week did you find most affirming and helpful?" (Brookfield, 2006)	In practice… Nothing… Discussion of problems given to us… The teaming up part to certificate the programs… The intensive exercises and the discussion in groups and pairs…
	"What action that anyone (teacher or student) took in class this week did you find most puzzling or confusing?" (Brookfield, 2006)	Nothing by now… Explanation about the *if* and *else*… Flowcharts are the most confusing for me… Every time… I didn't felt confused by anybody…
	"What about the class this week surprised you the most?" (Brookfield, 2006)	How easy the teacher explained… The *while* but I liked it… Each day I understand more… That while can be used in 4 ways… How in a short time with did a lot of programs…

Based on the majority of the expressions done by participants in the experimental group through the CIQ in those weeks in which aspects of the proposed approach were integrated it must be established that the null hypothesis must be rejected.

ResQ 2.3. How much time is needed to get confident with the different building blocks and structures while the understanding acquired according to the CIQ is improved using the proposed approach? (Mixed Approach)

> *$H_0$2.3:* The time needed to get confident with the different building blocks and structures will appear to be similar or higher and the understanding acquired as reflected in the CIQ will appear to be similar or lower while students learn with the proposed approach.
>
> *H_A2.3:* The time needed to get confident with the different building blocks and structures will appear to be reduced and the understanding acquired as reflected in the CIQ will appear to be higher while students learn with the proposed approach.

According to the results analyzed in ResQ 2.1 and ResQ 2.2 reflected in Tables 19, 20, 21, 22, 23, 24 and 25 it must be established that the null hypothesis *$H_0$2.3* must be retained. ResQ 2.2 established that through the CIQ the expressions of students participating were mostly positive in favor of the proposed approach as they were taught with it. However, the times associated with working with the proposed approach in classes appeared to be not necessarily affected in any way by it as established in ResQ 2.1.

ResQ 2.4. How much significant is the difference between the two groups of students in performance and how much positive is it in favor of the proposed approach? (Quantitative Approach)

The following are the hypotheses related to this question. As can be seen they are about the students' performance by means of the final score obtained in the classes comparing the control and the experimental groups.

$H_0$2.4: The means of the test scores for the two groups are not significantly different or the mean for the group working with the proposed approach is significantly lower.

H_A2.4: The mean test scores for the group working with the proposed approach is significantly higher.

In Table 26 and Table 27 the statistical results from question ResQ 1.1 are shown again. As established by the p level in 0.05 the null hypothesis must be retained.

Table 32. First Part of the Results Obtained After Running the Mann-Whitney U Test for the Class Scores (from Resq 1.1)

Ranks

	Group	N	Mean Rank	Sum of Ranks
FinalScore	Experimental	13	14.42	187.50
	Control	10	8.85	88.50
	Total	23		

As established for ResQ 1.1, the results obtained from the Mann-Whitney U test indicate that the null hypothesis *$H_0$2.4* must be retained based on $p < 0.05$. A statistically significant difference between the two groups does not exist.

Table 33. Second Part of the Results Obtained After Running the Mann-Whitney U Test for the Class Scores (from ResQ 1.1)

Test Statistics(b)

	FnalScore
Mann-Whitney U	33.500
Wilcoxon W	88.500
Z	-1.956
Asymp. Sig. (2-tailed)	.050
Exact Sig. [2*(1-tailed Sig.)]	.049(a)

a Not corrected for ties.
b Grouping Variable: Group

ResQ 2.5. How much significant is the difference between the two groups of students in motivation toward programming and how much positive is it in favor of the proposed approach? (Quantitative Approach)

The hypotheses related to this question are the following.

$H_0$1.2: Students using the proposed methodology were similarly motivated or less motivated toward using programming in comparison to those using the traditional approach.

H_A1.2: Students using the proposed methodology were more motivated toward using programming in comparison to those using the traditional approach.

Table 9, Table 10, Table 11 and Table 12 shows the data and results associated with this question and processed using the Mann-Whitney U test. As mentioned previously for ResQ 1.2 the results establish that for 9 of the 10 responses the null hypothesis must be retained, based on a significant level of $p < 0.05$. Thus, a statistically significant difference between the two groups was not found. Therefore, based on these results the null hypothesis must be retained.

ResQ 2.6. How much significant is the difference between the two groups of students in the predisposition toward continuing in the program and how much positive is it in favor of the proposed approach? (Quantitative Approach)

The hypotheses related to this question are the following.

> H_0*1.3:* Students using the proposed approach were similarly predisposed or less predisposed to continue in the program in comparison to those using the traditional approach.
>
> H_A*1.3:* Students using the proposed approach were more predisposed to continue in the program in comparison to those using the traditional approach.

Table 13, Table 14, Table 15 and Table 16 include the data and results related to ResQ 2.6 and ResQ 1.3. Table 16 has the final results after using the Mann-Whitney U test on the data. For ResQ 1.3 in 23 of the 25 items analyzed using the Mann-Whitney U test for comparison of the responses from both groups to the questionnaire, the test indicated that the null hypothesis must be retained. This is with a significant level of $p < 0.05$ establishing that there does not exist a statistically significant difference between the two groups.

Conclusion

This chapter presented the analysis of data and results obtained for the study based on a group of specific questions and hypotheses articulated to provide answers to two main questions. The questions are the following:

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

Based on the results presented, and analyzed statistically in most cases, for the sub questions and hypotheses associated with each main research questions it must be established that the specific response for ResQ 1 is that the proposed approach was efficient for teaching computer programming and for ResQ 2 is that the proposed approach did not provided better results in teaching computer programming.

For ResQ 1 this conclusion is based in the following specific ideas obtained from the results of the sub question and hypotheses related:

- There was not a difference in terms of performance between the students working with the proposed approach and those working with the traditional approach.
- There was not a difference in terms of the motivation of students toward programming between those working with the proposed approach and those working with the traditional approach.
- There was not a difference in terms of the predisposition of students toward continuing in the program between those working with the proposed approach and those working with the traditional approach.
- Students working with the proposed approach in general felt comfortable with it.

For ResQ 2 the concluding answer is based in the following specific ideas obtained from the results of the sub question and hypotheses related:

- There was not an improvement in the average time needed by students to get confident in programming using the main building blocks and structures required to be learned in the first programming class.
- Since the expressions made by students learning with the proposed approach were generally positive about their understanding of the concepts presented, it can be established that they benefit constantly from using it.
- The time needed by students to get confident with the different building blocks and structures while working with the proposed approach resulted to be similar with the time needed by students working with the traditional approach. A good or improved understanding of students did not represent any change in the time needed by students to get confident with the different building blocks and structures.
- There was not a significant difference in terms of performance between the students working with the proposed approach and those working with the traditional approach then it cannot be argued that any difference may be positively in favor of the proposed approach.
- There was not a significant difference in terms of the motivation of students toward programming between those working with the proposed approach and those working with the traditional approach then it cannot be argued that any difference may be positively in favor of the proposed approach.

- There was not a significant difference in terms of the predisposition of students toward continuing in the program between those working with the proposed approach and those working with the traditional approach. Therefore, it cannot be argued that any difference may be positively in favor of the proposed approach.

The analyses, results, and answers to the research questions and their sub questions and hypotheses will be discussed in detail in Chapter 5 including all the circumstances and implications of the study. These findings will be analyzed and interpreted based on the literature reviewed and the conceptual framework of the study as the main guide. Furthermore, Chapter 5 also includes the experiences of conducting this study and the implementation of the proposed approach, opinions, explanations about the phenomena, and the recommendations put forward from this investigation. Building on this study, the potential for future research is vast. Therefore, Chapter 5 includes a discussion with the purpose to generate new research ideas that can help improve the approach presented since it demonstrated to be useful within the context of the study.

CHAPTER 5. DISCUSSION, IMPLICATIONS, RECOMMENDATIONS

Introduction

This final chapter discusses the results presented in Chapter 4 in terms of their implications to this study, to the fields associated with the teaching of computer programming and the Information Technology (IT) body of knowledge. This chapter will also discuss directions for future research. The discussion of the results includes the details associated with the way the study addressed the problem presented through the proposed approach and recommendations on ways to enhanced it in future research. First of all, a summary of the results will be presented, followed by a detailed discussion of the results, how the conclusions relate to the review of literature, the limitations that affected the study, and the recommendations on the future work research. All of these aspects set the ground for concluding remarks of this dissertation document.

Summary of the Results

Research Problem

This study addressed the challenging process of teaching computer programming to freshmen students in computer sciences and other related programs. For students the learning of programming is very challenging. For faculty members the teaching of of computer programming implies the need to implement different approaches, strategies and innovative ways to teach their classes. This situation is addressed broadly in the literature through different studies that include a variety of approaches proposing different ways of improving this teaching process.

Purpose and Methodology of the Study

The purpose of this study was to examine if the implementation of a new approach for teaching computer programming to freshmen students in computer sciences causes an improvement in the acquisition of programming knowledge and skills, in comparison to students enrolled in a course that used the traditional approach. The study was done using a mixed method research design. Two groups were analyzed throughout the first semester of academic year 2012-2013 in an institution in Puerto Rico. The control group had been taught using the traditional approach of teaching programming, and the experimental group had been taught with the proposed approach.

Significance of the Study

Improving the way of teaching computer programming is important for IT related areas such as Computer Sciences, Information Systems, and Software Engineering, among others because by succeeding in this, the general process of teaching important concepts and skills may be shortened giving space in curriculum to integrate more advanced and important concepts. Also students may feel more prepared to address complex issues in programming. However, the most important effect considered was the possible reduction of the amount of students leaving the programs because they feel that computer programming is impossible or difficult to be learned.

Brief Details of the Existing Literature

The literature extensively addressed this situation. A significant amount of studies set out to find solutions or approaches that can be helpful in the improvement of the associated teaching process. An important number of studies were referred by Sheard, Simon, Hamilton, and Lönnberg (2009). Others presented their individual studies in a variety of related contexts, as occurred in the works presented by Bennedsen and Caspersen (2008), Blanco et al. (2009), Daly (2011), Esteves et al. (2011), Hawi (2010), Pokorny (2009), Thomas et al. (2011), and White (2006).

In a more recent investigation Dillon, Anderson, and Brown (2012) explored the use of three different programming environments as strategies to identify the one that may provide better results in teaching programming logic to novice students. The focus was on identifying how much beneficial an environment resulted depending on the assistive tools available. Furthermore, Wei (2013) referred to a study in which the effectiveness of using a methodology based on cooperative learning for teaching computer programming was evaluated. The results of comparing two groups demonstrated that the methodology used resulted as effective as the traditional methodology. However, students participating in the cooperative learning experience had a better perception of it in comparison to students not participating in it. The collaborative concept closely associated with the cooperative concept is one of the elements that was part of the proposed approach of this dissertation.

Vincenti, Braman, and Hilberg (2013) referred to a pilot study about the effectiveness of using what they named as reusable learning objects for teaching introductory programming courses. These objects are specifically designed resources directed to enhance the learning experience in classes and out of them, providing students with the possibility of having useful material to study while they are not in classes. These resources were evaluated as effective in supporting the general learning experience. Similarly, others like Sherman, Bassil, Lipman, Tuck, and Martin (2013), and Gaspar and Langevin (2012) referred to other resources and approaches directed to enhance the computer programming experience. As can be seen researchers continue looking for the ideal formula to work with the complex situation of teaching programming effectively. The general perception is that there is a lot more to do yet in this area.

Study Findings

The results associated with research question 1 (ResQ 1) established that the proposed approach resulted to be efficient for teaching computer programming to computer sciences freshmen students. This is based on the results of the sub questions and hypotheses associated which demonstrated that no difference existed between the two groups in terms of performance, motivation toward programming, and the predisposition of students to continue in the program. In general, student comments revealed they felt comfortable in learning with the proposed approach.

Regarding the results for research question 2 (ResQ 2) the evidence obtained from the study with this specific sample demonstrated that the proposed approach did not produce significantly better results in teaching computer programming to computer sciences freshmen students in comparison to those working with the traditional approach. The results from the sub questions and hypotheses related to it established that no difference existed between the two groups in the times needed by students to get confident with the main building blocks and structures required to learn in programming classes, and no difference existed between the two groups in terms of performance, motivation toward programming, and the predisposition of students to continue in the program. Students working with the proposed approach expressed constantly and in a positive way their understanding of the concepts presented.

Discussion of the Results

As established in the design of this study, two main research questions were formulated. Each of them has associated sub questions and hypotheses that are discussed in support of the two main research questions. Following is a discussion regarding the research questions:

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

According to the results obtained and referring to ResQ 1 the answer is that the proposed approach was efficient for teaching computer programming. The response is considered appropriate in the context of how the class using the proposed approach is compared to the class which used the traditional approach. It can be stated that both approaches were equally efficient for teaching computer programming to computer sciences freshmen students. In other words, the proposed approach resulted to be as efficient as the traditional approach in the referred teaching process. As stated, this conclusion is based on the results for the sub questions and hypotheses that are discussed in the following paragraphs.

ResQ 1.1. What is the difference between students working with the proposed approach in terms of performance as compared to the performance of the students that received instruction using the traditional approach? (Quantitative Approach)

The response to this question is that there is no significant difference between both groups in terms of performance. This is because the null hypothesis was retained establishing that the means of the scores obtained in the two groups were not significantly different. The experimental group in this case demonstrated the same level of competence demonstrated by the traditional group. This means that the topics covered throughout the semester in both groups were the same, and that they had the same level of difficulty. This can be concluded because the measure used to assess both classes was exactly the same. It was the one used to assess all the sections offered for the class used for the study.

ResQ 1.2. What is the difference between both groups on the motivation of students toward programming? (Quantitative Approach)

The response to this question is that there was no significant difference between both groups in terms of student's motivation toward programming. In answering the ten items of the survey questionnaire used to measure the motivation toward programming of all students the responses were consistent. The level of subjectivity that is part of the Modified Rosenberg Self-Esteem instrument was not enough to establish a difference between students in both groups. This may be considered evidence of the similarity between both groups and helps to confirm the results of ResQ 1.1 about the similarity in performance between them.

ResQ 1.3. What is the difference between both groups in the predisposition of students toward continuing in the program? (Quantitative Approach)

With question ResQ 1.3 the situation is very similar to the one occurred with question ResQ 1.2. There was no significant difference between both groups in terms of student's predisposition to continue in the program. The Ramalingham-Wiedenbeck Computer Self-

Efficacy Scale instrument has 25 items. Through those 25 items the responses of students were very consistent between the two groups leading again to consider this as evidence of the similarity existing between both groups on the last day of class. This also confirms the results of ResQ 1.1 about the similarity in performance between the two groups.

An important aspect of the used instrument is that self-efficacy, that may be very associated with motivation, can be considered a good predictor of the expectations students have about themselves. Students with high expectations are more probably sure of their plans for the future to continue doing what they understand they are doing well. This is the reason to consider the results from the instrument as a good predictor of student's predisposition to continue in the program, an important aspect of this study.

ResQ 1.4. How do the students feel in working with the proposed approach? (Qualitative Approach)

The answer for this question is that students felt comfortable in working with the proposed approach. The expressions were in general positive in favor of the proposed approach. In general, students' comments revealed they felt comfortable working with the proposed approach. This situation may be an explanation of the similarity found between experimental group and the control group in the sense that students that participated in the focus group probably never realized that they were working with an approach different from others. In fact, part of the expressions included that they felt being in a normal situation.

On the other hand, the expressions of being comfortable in classes while working with the proposed approach may be considered a sign of the efficiency of the approach in providing students the needed knowledge and skills for the course, at least in students' own perception.

They were probably convinced that the class was totally worked on the same circumstances of the other four sections offered during the semester, mostly when all students were evaluated with the same final test and in the same circumstances.

The concept of efficiency expressed in ResQ 1 is relative to the circumstances of considering the proposed approach as efficient as the traditional approach, as mentioned. Independently of how this efficiency is measured, it can be argued that the answers to the sub questions associated confirm it. The absence of differences between both groups in terms of performance, motivation toward programming, and in the predisposition to continue in the program establish that the implementation of the different approaches in both groups was not a factor in promoting a significantly different experience for students.

Within the specific context of the study, the initial expectation was to find a significant difference between both groups in all three aspects as referred in ResQ 1.1, ResQ 1.2, and ResQ 1.3. A greater difference was expected to occur when comparing the means of the scores for both groups and significant differences were expected to occur with the results compared for the instruments used. Some aspects may explain this situation. One can be associated with the small number of students that finally participated in the study. Another one can be the moderate implementation of the proposed approach. A third one is associated with aspects not considered in the design of the study that are finally identified as important. For example, an aspect not considered is the students that dropped or abandoned the classes in both groups. Another one is the opinion of students working with the traditional approach.

Considering these situations as limitations that can be affecting the results of the study, some of their possible impact is described below:

- The small number of participants finally considered for the study is significantly less than the original amount of participants established in the design. In research it is generally established that having small amounts of participants is adverse and may affect the validity of a study.
- In implementing the proposed approach, a decision was made to put into practice the process moderately. It was done without moving it to the full potential of the original idea. The reason was to assure they developed the knowledge and skills needed for the semester without overwhelming them. The effect was that both groups at the end had a very similar experience but presented by means of a different approach which they perceived as valid.
- Aspects not considered for the study may have more or less impact depending on the circumstances. An aspect that is suspected to be important for the specific study circumstances is the students that dropped or abandoned the class in both the experimental group and the control group. The reasons behind their decision to leave the classes may be important to be considered in the evaluation of the effectiveness of either of the approaches.
- Another aspect not considered that may be of importance but probably less than the described above was the opinion of students working with the traditional approach. Their comfort with the traditional approach may be significant in determining which approach is considered better.

It is possible that the aforementioned limitations did not affect the obtained results; however, this must be addressed in future investigations. The results of this study show that the proposed approach worked. It was as appropriate for teaching computer programming as the traditional approach. Therefore, it is safe to say that the proposed approach expands the discussion of teaching approaches regarding computer programming.

The discussion associated with research question 2 (ResQ 2) follows.

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

Based on the majority of the results that establish the answers to the questions associated with ResQ 2, its answer is that the proposed approach did not provided better results in teaching computer programming, at least to a significant level. The referred results are from the sub questions and hypotheses that will be discussed in the following paragraphs.

ResQ 2.1. What is the improvement, if any, in the average time needed by students to get confident in programming using the main building blocks and structures required to be learned in the first programming class? (Quantitative Approach)

After calculating and analyzing the average time needed by students to be confident in programming using the main building blocks and structures required in the first programming class, it was determined that an improvement is not observed based on the data obtained from the classes corresponding to both the experimental and the control group. This was established after retaining the null hypothesis associated with this question in which a similarity was found in terms of the average times referred. There was no significant difference between both average times. The average time of the experimental group was higher but not enough in order to establish a real difference.

The result of this question was particularly affected by one important aspect discussed to explain the answers associated with ResQ 1. It is the one related to the moderate implementation of the proposed approach. This circumstance is directly related to the situation of not having a real difference between the average times obtained from both groups. As presented in Table 28, the individual times obtained for the different main areas have differences between them in some cases; however, all of them occurred within almost the same frame of time. The benefit of having completed the *loops* area earlier was against to having completed *input* and *selection control* later. This had the effect of having all the areas presented at the end of the semester without any explicit difference between the groups.

Table 34. Main Areas Covered and the Average Weeks in Which They Were Discussed for Both Groups under Study (from Table 18)

Areas	Control	Experimental
Analysis and Design	5.00	4.00
Basics	5.00	5.67
Output	5.00	5.00
Input	5.00	10.00
Selection Control	6.67	12.33
Loops	11.00	7.75
Methods	16.00	17.00
Files	12.00	16.00

ResQ 2.2. Based on the Critical Incident Questionnaire (CIQ) instrument for formative assessment, how much do the computer science freshmen students participating in the study

using the proposed approach to teaching programming benefit constantly from using it? (Qualitative Approach)

The analysis of the data obtained from the CIQ responses led to conclude that students' positive comments regarding their experience with the proposed approach reveals they developed an understanding of the concepts presented. This answer came from a rejection to the null hypothesis associated with this question. As occurred with the focus group analyzed in ResQ 1.4, the expressions made revealed an acceptance of the approach used as appropriate for the teaching process according to the perception of students. The results of both questions (ResQ 1.4 and ResQ 2.2) are helpful in supporting each other, in favor of the proposed approach. As established previously, in the quantitative results discussed to this point the proposed approach demonstrated being similar to the traditional approach in terms of efficiency. This explains the reasons behind the positive expressions made by students through the CIQ. These expressions were more important because all the presented as part of the answer were exclusively from the CIQ submissions corresponding to the weeks in which elements of the proposed approach were implemented.

ResQ 2.3. How much time is needed to become confident with the different building blocks and structures while the understanding acquired according to the CIQ is improved using the proposed approach? (Mixed Approach)

In combining the answers from ResQ 2.1 and ResQ 2.2 it was established that the null hypothesis was retained. The expressions made by students presented a positive view of the proposed approach demonstrating they felt comfortable and possibly with confidence about their understanding of the concepts worked. However, the difference in the average times needed to be confident with them was not significant leading to retain the null hypothesis. These results help to confirm all the findings discussed to this point establishing one of the instances in which the qualitative and the quantitative results obtained allow to explain the phenomena. Here the expressions from the CIQ helped to confirm the situation of the proposed approach that is similar to the situation of the traditional approach, but without allowing establishing that the first is significantly better than the other.

ResQ 2.4. How much significant is the difference between the two groups of students in performance and how much positive is it in favor of the proposed approach? (Quantitative Approach)

As discussed, for ResQ 1, the means of the test scores for the two groups were not significantly different, and then the answer to this question is that the difference between the two groups of students in performance was not significant and the difference was only a little bit positive in favor of the proposed approach, after realizing that the null hypothesis must be retained. Since this answer is directly related to a corresponding question that is part of ResQ 1, it supports all the findings discussed to this point, and the explanations to this behavior are the same already argued and explained in more detail in the following sections.

ResQ 2.5. How much significant is the difference between the two groups of students in motivation toward programming and how much positive is it in favor of the proposed approach? (Quantitative Approach)

As discussed, for ResQ 1, students using the proposed methodology were similarly motivated toward using programming in comparison to those using the traditional approach. The results of the Modified Rosenberg Self-Esteem instrument for the two groups were not significantly different, and then the answer to this question is that the difference between the two groups of students in motivation toward programming was not significant and the difference was only a little bit positive in favor of the proposed approach, after realizing that the null hypothesis must be retained. Since this answer is directly related to a corresponding question that is part of ResQ 1, it supports all the findings discussed to this point. This behavior is argued and explained in more detail in the following sections.

ResQ 2.6. How much significant is the difference between the two groups of students in the predisposition toward continuing in the program and how much positive is it in favor of the proposed approach? (Quantitative Approach)

As discussed, for ResQ 1, students using the proposed methodology were similarly predisposed to continue in the program in comparison to those using the traditional approach. The results of the Ramalingham-Wiedenbeck Computer Self-Efficacy Scale instrument for the two groups were not significantly different, and then the answer to this question is that the difference between the two groups of students in the predisposition toward continuing in the program was not significant and the difference was only a little bit positive in favor of the proposed approach, after realizing that the null hypothesis must be retained. Since this answer is directly related to a corresponding question that is part of ResQ 1, it supports all the findings discussed to this point and the explanations to this behavior are the same already argued and explained in more detail in the following sections.

Implications of the Study Results

The results of this study have the main implication of providing a new approach to researchers in computer programming and teaching. The results demonstrated that the proposed approach is as efficient as the traditional approach. Participants' comments revealed the proposed approach was appropriate for their learning of computer programming. Their comments were supported with valid results from instruments on motivation and self-efficacy making these results stronger. The area of teaching computer programming, the IT field, and other related areas have a new line of research, testing, and implementation that can be explored in a variety of contexts and situations. Now the concept of immersion obtained mainly for the field of spoken languages as a strategy for teaching can be broadened to IT and other related areas in contexts associated with programming and possibly with other aspects of computing.

Participant's acceptance of the proposed approach surfaces new concepts for analysis. One is the programming language immersion that explains the process of exposing students to code and leading them to analyze and interpret statements from an already worked program in order to determine how it works, as defined in Chapter 1. The other is programming topic immersion, also defined in Chapter 1, which implies working earlier in a semester with topics that are more complex than those normally taught in a moment, requiring students to analyze them. Through these concepts, the idea of programming immersion originally presented in Harper (2006) was brought to a different level of implementation and formality. The concept of intensive exercises, which emerged in the process of implementing the proposed approach, can be developed and considered among the strategies associated with extreme programming and agile software development.

As seen, the IT education area is probably the most related to the ideas worked in implementing the proposed approach and the concepts analyzed through the study. Therefore, the field in general can explore them in other contexts, such as professional training activities for organizations with IT employees or training to employees in new tools or technologies.

Limitations

The following are the main aspects identified as limitations of the study.

- The limitation about the small number of students was discussed in the methodology section of the study. Twenty participants were expected for each group. It was argued that the effect of such amount of participants was minimized through the use of the mixed method research methodology. This was because the results obtained statistically can be confirmed with the comments made by students in different instances. This situation occurred after applying the triangulation methodology. Also students' comments are supported with the use of a statistical test that is accepted for comparison of small samples (the Mann Whitney U test). However, this situation is a concern that may be important to consider in any replication of this study or any other future similar research.
- The proposed approach was not implemented to its full potential. Instead, a moderate implementation of this new educational strategy served to maintain a safe and comfortable teaching learning experience for the students in the following ways:
    - They should not be overwhelmed with more material than the one they can manage safely and feel comfortable working with.
    - In addition, they should not be overwhelmed with strategies that can cause a sensation of being lost, unstable or uncomfortable.

  - The material established in the syllabus for the course must be completed without risk of leaving any topic out or of diminishing any of them.
- The original idea of immersion was conceived with the possibility of including important material offered normally in the second part of the course. These two important areas to include in the immersion exercises are working with arrays and object oriented programming through the definition of simple classes and objects. Working the immersion process including these areas was impossible in the context of the implementation done for the study. This implies a direct effect on the study results in terms of not having a difference between the two groups in reflecting an improvement in the times needed to teach the most important building blocks and structures of computer programming.
- As mentioned, some aspects were not considered for the study that may be of importance. The lack of considering the students that dropped or abandoned the classes is important and can have an effect on the results of the study. After conducting the study, it was realized that this aspect was probably significant in the results. Table 29 presents the situation of the samples for the study with the amount of students that dropped or abandoned the classes.

Table 35. Final Samples for the Study

	Experimental Group	Control Group
Enrolled students	19	19
Signed Consent Form	18	12
Dropped from class or disappeared	5	6
Dropped or disappeared who signed	5	2
Completed the final surveys	13	10
Completed posttest who signed	13	10

- Table 29 depicts the students that accepted to be part of the study. This table shows that five of them left the experimental group and two dropped from the control group. In most of these cases, the reasons for dropping the course are unknown. It is uncertain if some of them left the classes because of any situation associated to any of the approach used in teaching them. However, in the experimental group two students expressed the first day of class that they really did not want to be in the Computer Sciences program. They initially accepted in order to have the opportunity to request a change to the program they really wanted. In either of these cases it may be really important to consider these students' situations, opinions and performance as part of a future study. The impact of these aspects should be part of the design of a future study with similar characteristics.
- The other aspect identified was not obtaining the opinion through a focus group of students in the control group working with the traditional approach. The opinion of these students may be important in confirming if they were comfortable with the class

while working with the traditional approach. A study of this nature would allow comparing qualitatively the aspects analyzed in this study between both groups. In this study, a focus group was done only with students that were in the experimental group working with the proposed approach. This exercise resulted to be very interesting and useful in providing information about the participants feelings regarding the proposed approach throughout the semester. It helped clearly to support the conclusion that the proposed approach resulted to be efficient in the process of teaching computer programming.

Recommendations for Further Research or Intervention

This study contributes to building knowledge about the challenging process of teaching computer programming to freshman students in computer sciences and other related areas. However, many more interesting research questions could be answered by using the proposed approach in other contexts. The most important recommendations for future research are listed as follows.

- The study can be replicated by increasing the sample. One way is repeating the study in similar circumstances assuring a maximum group of 20 participants, as defined in the design of this study. Another way is by redefining the methodology and repeating it with a maximum of 40 students for each sample, where 40 of them are enrolled in two sections associated with the control group, and the other 40 are enrolled in two sections associated with the experimental group. In other words, the samples for both the control and the experimental groups can be of 40 students by including 2 class groups of 20 in each, having a maximum of 80 participants. More groups can be added to the study considering their availability on the site where the study is developed.

- The study can be replicated by including in the immersion activities developed earlier in the course topics usually offered at the end of it. Some areas to consider are working with methods and working with files. Both areas have the potential to be integrated very early in the semester. The constant exposure to the use of methods and files integrated as part of the immersion exercises on the repetitions and selection control structures may be excellent in developing the skills of using them effectively.
- The study can also be replicated by including in the immersion activities topics offered in the second semester of the course. Some areas to consider in this implementation are working with arrays and object oriented programming through the definition of simple classes and objects.
- The study can be replicated, with the necessary adjustments, assuring the inclusion of all the data and the comments of those students that opt to drop from the courses or simply leave them. The inclusion of this data would require an adjustment in the methodology. Also, it would be very important to elicit comments that would help to determine if the reasons for leaving the classes are related to the approach used in teaching the courses or are associated with other unrelated situations.
- The study can be replicated including a focus group also to obtain the opinions of the control group. Focusing on the two groups of students- the control group and the experimental group- calls for a qualitative comparison of the comments revealed through the focus groups settings between these two groups.

- The study, with all or part of the previous recommendations integrated, can be worked in multiple sites in order to have samples that reflect a greater diversity, thereby helping to improve its validity.
- The proposed approach can be considered for implementation in other IT courses associated with programming. It is understood that most of the courses in IT related areas are appropriate for future implementation of the approach.
- The proposed approach can be considered in other areas not related to computer programming. It is understood that it can be appropriate in teaching courses related to different technological areas different from computer programming. Thus, the proposed approach is not limited to computing or IT.
- The proposed approach can be explored as a strategy for implementation within the context of organizations for trainings and other related activities.

Conclusion

The teaching of computer programming to freshmen students in IT and other related areas, such as Computer Sciences, was described in this study as a complex activity. This study presented ideas that are considered useful. Two research questions were established in order to investigate if the ideas presented, organized as the concept of a new approach for teaching computer programming, are appropriate as a solution to the established problem. The research questions were the following:

ResQ 1

How efficient is the proposed approach for teaching computer programming to computer sciences freshmen students? (Mixed Approach)

ResQ 2

How does the proposed approach provide better results in teaching computer programming to computer sciences freshmen students? (Mixed Approach)

The specific answer to ResQ 1 is that the proposed approach was efficient for teaching computer programming, and for ResQ 2 it is that the proposed approach did not provided better results in teaching computer programming. These answers were broadly discussed based on the results obtained through analyzing a group of sub questions and hypothesis worked through an experiment with two groups of students that were compared to determine if the proposed approach is better than the traditional approach used in teaching programming.

The results were reported and discussed. The explanations for these results explored the reasons, limitations, and possible ways to improve through future studies. Of significant importance in this research process is that new ideas are generated and possibilities for development, inclusively in the specific context of the study presented. The concepts of programming language immersion, programming topics immersion, and intensive exercises are there, ready for their development supported by all the important strategies, ideas and educational theories that helped develop them as part of the proposed approach. One of the supporting strategies came from extreme programming, however, everything was backed up with the following important concepts and theories from education: collaborative learning, problem based learning, self-directed learning, experiential learning and transformative learning.

REFERENCES

Abdullah, M. Z., Idris, S., & Subramaniam, N. K. (2006). Implementing virtual pair programming in E-learning environment. *Journal of Information Systems Education, 17*(2), 113-117. Retrieved from http://search.proquest.com/docview/200164893?accountid=27965

Agarwal, R. & Umphress, D. (2008). Extreme programming for a single person team. *Proceedings of the 46th Annual Southeast Regional Conference on XX,* 28-29 March 2008 (pp. 82-87). Auburn, AL: Association for Computer Machinery, Inc.

Ali, S. (2005). Effective teaching pedagogies for undergraduate computer science. *Mathematics and Computer Education, 39*(3), 243-257. Retrieved from ProQuest Education Journals.

Askar, P., & Davenport, D. (2009). An investigation of factors related to self-efficacy for Java programming among engineering students. *Turkish Online Journal of Educational Technology, 8*(1). Retrieved from http://www.tojet.net/volumes/v8i1.pdf#page=27

Association for Computer Machinery. (2005). *Computing Curricula 2005: The overview report.* Retrieve from http://www.acm.org/education/

Bandura, A. (1977) Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, *84*(2), 191-215.

Becker, B. W. (2001). Teaching CS1 with karel the robot in Java. In H. Walker, R. McCauley, J. Gersting, & I. Rusell (Eds.), *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education,* 1 February 2001 (pp. 50-54). Charlotte, NC: Association for Computer Machinery, Inc.

Bennedsen, J., & Caspersen, M. E. (2008). Optimists have more fun, but do they learn better? On the influence of emotional and social factors on learning introductory computer science. *Computer Science Education, 18*(1), 1 – 16.

Bergin, J. (2006). ITICSE '06 Proceedings of the 11th Annual SIGCSE Conference on innovation and Technology in Computer Science Education. In R. Davoli, M. Goldweber, & P. Salomoni (Ed.), *Karel universe drag & drop editor* (p. 307). New York, NY: Association for Computer Machinery, Inc.

Bergin, S., & Reilly, R. (2005). The influence of motivation and comfort-level on learning to program. *Proceedings of the 17th Workshop on Psychology of Programming, PPIG '05,* 293 – 304.

Bergin, S., & Reilly, R. (2006). Predicting introductory programming performance: A multi-institutional multivariate study. *Computer Science Education, 16*(4), 303-323. doi: 10.1080/08993400600997096

Blanco, J., Losano, L., Aguirre, N., Novaira, M. M., Permigiani, S., & Scilingo, G. (2009). An introductory course on programming based on formal specification and program calculation. *Special Interest Group on Computer Science Education Bulletin, 41*(2), 31-37. doi: 10.1145/1595453.1595459

Borge, R., Fjuk, A., & Groven, A.K. (2004). Using Karel J collaboratively to facilitate object-oriented learning. *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, 580- 584. Retrieve from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1357481&isnumber=29792

Bosson, J.K., Swann, W.B.J., & Pennebaker, J.W. (2000). Stalking the perfect measure of implicit self-esteem: The blind men and the elephant revisited? *Journal of Personality and Social Psychology, 79*(4), 631–643.

Brookfield, S. D. (1995). *Becoming a critically reflective teacher*. San Francisco, CA: Jossey-Bass.

Brookfield, S. D. (2006). *The skillful teacher: On technique, trust, and responsiveness in the classroom (2nd ed.).* San Francisco, CA: Jossey-Bass.

Cantwell-Wilson, B. & Shrock, S. (2001). Contributing to success in an introductory computer science course: A study of twelve factors. *Special Interest Group on Computer Science Education Bulletin, 33*(1), 184-188. doi: 10.1145/366413.364581

Cheng, L., Li, M., Kirby, J. R., Qiang, H., & Wade-Woolley, L. (2010). English language immersion and students' academic achievement in English, Chinese and mathematics. *Evaluation & Research in Education, 23*(3), 151-169.

Cheung, J. C., Ngai, G., Chan, S. C., & Lau, W. W. (2009). Filling the gap in programming instruction: a text-enhanced graphical programming environment for junior high students. In S. Fitzgerald, M. Guzdial, G. Lewandowski, & S. Wolfman (Eds.), *SIGCSE'09 Proceedings of the 40th ACM Technical Symposium on Computer Science Education,* 3-7 March 2009 (pp. 276-280). Chattanooga, TN: Association for Computer Machinery, Inc.

Cilliers, C., Calitz, A., & Greyling, J. (2005). The effect of integrating an iconic programming notation into CS1. In J. Cunha, W. Fleischman, V. K. Proulx, & J. Lourenço (Eds.), *ITICSE'05 Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education,* 27-29 June 2005 (pp. 108-112). Monte de Caparica, Portugal: Association for Computer Machinery, Inc.

Cliburn, D.C. (2008). Student opinions of Alice in CS1. *Frontiers in Education Conference,*T3B-1-T3B-6. Retrieve from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4720254&isnumber=4720249

Cobb, B., Vega, D., & Kronauge, C. (2006). Effects of an elementary dual language immersion school program on junior high achievement. *Middle Grades Research Journal, 1*(1), 27-47.

Conway, M., Audia, S., Burnette, T., Cosgrove, D., & Christiansen, K. (2000). Alice: lessons learned from building a 3D system for novices. In T Turner, & G. Szwillus (Eds.), *CHI'00 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1-6 April 2000 (pp. 486-493). Amsterdam, Netherlands: Association for Computer Machinery, Inc.

Cooper, C. R., & Schindler, P. S. (2011). *Business research methods (11th ed.).* Boston, MA: McGraw-Hill.

Creswell, J., & Plano Clark, V. L. (2010). *Designing and conducting mixed methods research (2nd ed.).* Thousand Oaks, CA: Sage.

Daly, T. (2011). Minimizing to maximize: An initial attempt at teaching introductory programming using Alice. *Journal Computing Sciences in Colleges, 26*(5), 23-30.

de Kereki, I. F. (2008). Scratch: Applications in Computer Science 1. *Frontiers in Education Conference,* T3B-7-T3B-11.

DePasquale, P. (2003). In Lee J. (Ed.), *Implications on the learning of programming through the implementation of subsets in program development environments*. United States -- Virginia: Virginia Polytechnic Institute and State University. Retrieved from http://search.proquest.com/docview/305299975?accountid=27965

Dillon, E., Anderson, M., & Brown, M. (2012). Comparing feature assistance between programming environments and their "effect" on novice programmers. *Journal Computer Sciences in Colleges, 27*(5), 69-77.

Erickson, J., Lyytinen, K., & Siau, K. (2005). Agile modeling, agile software development, and extreme programming: The state of research. *Journal of Database Management, 16*(4), 88-100. doi: 10.4018/jdm.2005100105

Esteves, M., Fonseca, B., Morgado, L., & Martins, P. (2011). Improving teaching and learning of computer programming through the use of the Second Life virtual world. *British Journal of Educational Technology*, *42*(4), 624-637. doi: 10.1111/j.1467-8535.2010.01056.x

Fagerland, M., Sandvik, L., & Mowinckel, P. (2011). Parametric methods outperformed non-parametric methods in comparisons of discrete numerical variables. *BMC Medical Research Methodology, 11*(1), 44-51. doi: 10.1186/1471-2288-11-44

Fay, M. P., & Proschan, M. A. (2010). Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys, 4*(2010), 1-39.

Fernandez-Aleman, J. L. (2010). Automated assessment in a programming tools course. *IEEE Transactions on Education, PP*(99), 1-6.

Gaspar, A, & Langevin, S. (2012). An experience report on improving constructive alignment in an introduction to programming. *Journal Computer Sciences in Colleges, 28*(2), 132-140. Retrieved from ACM Digital Library.

Goel, S., & Kathuria, V. (2010). A novel approach for collaborative pair programming. *Journal of Information Technology Education*, *9,* 183-196. Retrieved from http://www.editlib.org/p/111363.

Govender, I. (2009). The learning context: Influence on learning to program. *Computing Education, 53*(4), 1218-1230.

Grow, G. (1994). In defense of the staged self-directed learning model. *Adult Education Quarterly, 44*(2), 109-14. Retrieved from http://alec2.tamu.edu/grad_courses/611/modules/Module2/Lesson2/Grow02.PDF

Grow, G. O. (1991). Teaching learners to be self-directed. *Adult Education Quarterly, 41*(3), 125-49. doi: 10.1177/0001848191041003001

Hadar, I., Sherman, S., & Hazzan, O. (2008). Learning human aspects of collaborative software development. *Journal of Information Systems Education, 19*(3), 311-319. Retrieved from http://search.proquest.com/docview/200119144?accountid=27965

Harper, S. (2006). Immersion language theory meets CS. *Journal of Computing in Small Colleges, 22*(2), 85-91.

Hawi, N. (2010). Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course. *Computing Education, 54*(4), 1127-1136. doi: http://dx.doi.org/10.1016/j.bbr.2011.03.031

Hays, W. L. (1973). *Statistics for the Social Sciences (2nd ed.).* New York, NY: Holt, Rinehart, and Winston.

Henriksen, P. & Kölling, M. (2004). Greenfoot: Combining object visualisation with interaction. In J. Vlissides, D. Schmidt, & G. Cohen (Eds.), *OOPSLA '04 Companion to the 19th annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications,* 24-28 May 2004 (pp. 73-82). British Columbia, Canada: Association for Computer Machinery, Inc.

Hernandez Sampieri, R., Fernandez Collado, C., & Baptista Lucio, P. (2003). *Metodología de la investigación (2nd ed.).* Mexico: McGraw-Interamericana.

Hoffer, J. A., George, J. F., & Valacich, J. S. (2011). *Modern systems analysis & design (6th ed.).* Boston, MA: Prentice Hall.

Johnsgard, K., & McDonald, J. (2008). Using Alice in overview courses to improve success rates in Programming I. *IEEE 21st Conference on Software Engineering Education and Training,* 129-136. Retrieve from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4556958&isnumber=4556933

Johnson, R. B. Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of Mixed Methods Research, 1*(2), 112-133.

Kobayashi, O., Kawabata, M., Sakai, M., & Parkinson, E. (2006). Analysis of the interaction between practices for introducing XP effectively. *ICSE'06 Proceedings of the 28th international Conference on Software Engineering,* 20-28 May 2009 (pp. 544-550). Shanghai, China: Association for Computer Machinery, Inc.

Knowles, M., Holton, E., & Swanson, R. (2011). *The adult learner: The definitive classic in adult education and human resource development (7th ed.).* Burlington, MA: Elsevier/Butterworth-Heinemann.

Kranch, D. A. (2010). *A study of three instructional sequences for developing computer programming expertise in novice learners*. (Doctoral dissertation). Capella University. *ProQuest Dissertations and Theses.* Retrieved from http://search.proquest.com/docview/275978951?accountid=27965

Larason, K. (1995). Using Karel the robot as a classroom motivator. *3C ON-LINE, 2*(4), 6. doi: 10.1145/216922.216927

Mangalaraj, G., Mahapatra, R., & Nerur, S. (2009). Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems, 18*(4), 344-354. doi: 10.1057/ejis.2009.23

Merriam, S. B., Caffarella, R. S., & Baumgartner, L. M. (2007). *Learning in adulthood a comprehensive guide (3rd ed.).* San Francisco, CA: John Wiley & Sons.

McKeachie, W. J. & Svinicki, M. (2006). *McKeachie's teaching tips: Strategies, research, and theory for college and university teachers (12th ed.)*. Belmont, CA: Wadsworth Cengage Learning.

Onwuegbuzie, A.J., & Teddlie, C. (2003). A framework for analyzing data in mixed methods research. In A. Tashakkori & C. Teddlie (Eds.), *Handbook of mixed methods in social and behavioral research* (pp. 351–383). Thousand Oaks, CA: Sage.

Othman, M., & Othman, M. (2012). The proposed model of collaborative virtual learning environment for introductory programming course. *Turkish Online Journal of Distance Education*, *13*(1), 100-111.

Overmars, M. (2004). Teaching computer science through game design. *Computer*, *37*(4), 81-83. Retrieve from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1297314&isnumber=28844

Palloff, R. M. & Pratt, K. (2007). *Building online learning communities: Effective strategies for the virtual classroom (2nd ed.).* San Francisco, CA: John Wiley & Sons, Inc.

Patfis, R. E. (1981). *Karel the Robot: A gentle introduction to the art of programming,* John Wiley & Sons.

Platt, J. (2012). A Mexico City-based immersion education program: Training mental health clinicians for practice with Latino communities. *Journal of Marital & Family Therapy, 38*(2), 352-364.

Pokorny, K. L. (2009). Introduction to computing: A fresh breadth of disciplines. *Journal of Computing Sciences in Colleges, 24*(5), 166-172. Retrieved from http://dl.acm.org/citation.cfm?id=1516630

Qiang, H., & Kang, Y. (2011). English immersion in China as a case of educational transfer. *Frontiers of Education in China, 6*(1), 8-36.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research, 19*(4), 367-81. doi: 10.2190/C670-Y3C8-LTJ1-CT3P

Ritchey, F.J. (2002). *Estadística para las Ciencias Sociales: El potencial de la imaginación estadística.* Mexico City, Mexico: McGraw-Hill Interamericana.

Sanders, D., & Dorn, B. (2003). Classroom experience with Jeroo. *Jornal of Computing Sciences in Colleges, 18*(4), 308-316. Retrieved from http://dl.acm.org/citation.cfm?id=767644

Schmitt, D.P., & Allik, J. (2005). Simultaneous administration of the Rosenberg self-esteem scale in 53 nations: Exploring the universal and culture-specific features of global self-esteem. *Journal of Personality and Social Psychology, 89*(4), 623–642.

Schön, D. A. (1987). *Educating the reflective practitioner. Toward a new design for teaching and learning in the professions.* San Francisco, CA: The Jossey-Bass Higher Education Series.

Sheard, J., Simon, S., Hamilton, M., & Lönnberg, J. (2009). Analysis of research into the teaching and learning of programming. In M. Clancy, M. Caspersen, & R. Lister (Eds.), *ICER'09 International Computing Education Research Workshop,* 10-11 August 2011 (pp. 93-104). Berkley, CA: Association for Computer Machinery, Inc.

Sheffer, C. E., Barone, C., & Anders, M. E. (2011). Training nurses in the treatment of tobacco use and dependence: Pre- and post-training results. *Journal of Advanced Nursing, 67*(1), 176-183.

Sherman, M., Bassil, S., Lipman, D., Tuck, N., & Martin, F. (2013). Impact of auto-grading on an introductory computing course. *Journal Computer Sciences in Colleges, 28*(6), 69-75. Retrieved from http://dl.acm.org/citation.cfm?id=2460171

Silverstone, P.H., & Dadashova, R. (2012). Atomoxetine treatment for nicotine withdrawal: A pilot double-blind, placebo-controlled, fixed-dose study in adult smokers. (2012). *Annals of General Psychiatry, 11*(1), 6-16.

Sprankle, M. (2005). *Problem solving and programming concepts (7th ed.).* Edition: Prentice Hall.

Swanson, R. A., & Holton, E. F., III. (Eds.). (2005). *Research in organizations: Foundations and methods of inquiry*. San Francisco, CA: Berrett-Koehler.

Tashakkori, A., & Creswell, J.W. (2007). Exploring the nature of research questions in mixed methods research. *Journal of Mixed Methods Research, 1*(3), 207-211.

Taylor, E. W. (2000). Fostering Mezirow's transformative learning theory in the adult education classroom: A critical review. *Canadian Journal for the Study of Adult Education*, *14*(2), 1-28.

Thomas, M. K., Ge, X., & Greene, B. A. (2011). Fostering 21st century skill development by engaging students in authentic game design projects in a high school computer programming class. *Journal of Educational Computing Research*, *44*(4), 391-408.

Trochim, W. M. (2006). *The research methods knowledge base (2nd ed.).* Retrieve from http://www.socialresearchmethods.net

Truong, N., Roe, P., & Bancroft, P. (2005). Automated feedback for "fill in the gap" programming exercises. *Proceedings of the 7th Australasian Conference on Computing Education - Volume 42,106,* 117-126.

Usborne, E., Peck, J., Smith, D., & Taylor, D. M. (2011). Learning through an aboriginal language: The impact on students' English and aboriginal language skills. *Canadian Journal of Education, 34*(4), 200-215.

Vincenti, G., Braman, J., & Hilberg, J. S. (2013). Teaching introductory programming through reusable learning objects: a pilot study. *Journal Computer Sciences in Colleges, 28*(3), 38-45. Retrieved from http://dl.acm.org/citation.cfm?id=2400172

Warfield, D. (2010). IS/IT research: A research methodologies review. *Journal of Theoretical & Applied Information Technology, 13*(1/2), 28-25. Retrieved from EBSCOhost.

Wei, D. (2013). An evaluation of a cooperative learning method in Programming and Problem Solving I. *Journal Computer Sciences in Colleges, 28*(3), 69-77. Retrieved from ACM Digital Library

Wesely, P. M. (2010). Language learning motivation in early adolescents: Using mixed methods research to explore contradiction. *Journal of Mixed Methods Research, 4*(4), 295-312.

White, G. (2006). Visual Basic programming impact on cognitive development of college students. *Journal of Information Systems Education, 17*(4), 421-427. Retrieved from ABI/INFORM Global. (Document ID: 1211107131).

Wiedenbeck, S., LaBelle, D., & Kain, V.N.R. (2004). Factors affecting course outcomes in introductory programming. In E. Dunican, & T. R. G. Green (Eds.), *PPIG '04 Proceedings in 16th Workshopof the Psychology of Programming Interest Group,* April 2004 (pp. 97-110). Carlow, Ireland: Association for Computer Machinery, Inc.

Wlodkowski, R. J. (2008). *Enhancing adult motivation to learn: A comprehensive guide for teaching all adults (3rd ed.)*. San Francisco, CA: Jossey-Bass.

Yamauchi, L. A., Lau-Smith, J., & Luning, R. I. (2008). Family involvement in a hawaiian language immersion program. *School Community Journal, 18*(1), 39-60.

APPENDIX. INSTRUMENT USED IN THE STUDY

Focus Group Protocol

Introductory script:
The purpose of this focus group is to know your feelings and motivations associated with the course. You were cited in this date in order to assure you that your expressions will not have any effect in your final grade. You can feel free to make any type of expressions positive or negative about your personal experience with the course. It is important to have your sincere opinion in order to obtain a real picture of the effectiveness achieved with the approach used in teaching the course. Your expressions will not be in any way offensive to the researcher because they are considered an essential part of the study. The focus group will take an hour. However, if you are interested in extending the discussion for an extra half hour please express this in the last few minutes of the activity, when the researcher indicate that the time is gone. Now we will begin the focus group through presenting an initial question that will be followed by some others. You can present also any question you want.
Questions for the focus group:
These questions will be used as a general guide. They can be changed or omitted as need based on the discussion developed throughout the focus group and the time restrictions established. It is expected that all participants have the same opportunity to answer, but they are free to opt for maintain in silence.

- What do you think about your learning experience in the class?
- How can you describe the teaching-learning process occurred throughout the class?
- What do you think about the effectiveness of the approach used in teaching the class?
- How much comfortable do you had felt during the course with the approach used?
- What is the more positive comment you can make about the class, if any?
- What is the more negative comment you can make about the class, if any?
- Do you have any recommendation for improvement in the approach used to teach the class?