An approach for the use of Learning Objects in teaching Computer Programming concepts

Luiz Ricardo Begosso
Fundação Educacional do Município de Assis
Centro de Pesquisas em Informática
Assis, Brazil
begosso@femanet.com.br

Luiz Carlos Begosso
Fundação Educacional do Município de Assis
Faculdade de Tecnologia de Assis
Assis, Brazil
lbegosso@femanet.com.br

Raíssa Helena Begosso
BS Tecnologia
São Paulo, Brazil
raissabegosso@gmail.com

Abstract - **The teaching of algorithms and programming concepts to students of the first years of Computer Science course has been a major challenge, because students often have difficulty understanding the subject, leading to a high dropout rate. At our institution, dropout rate in the first year of the Computer Science course is 26%, which led us to think of pedagogical strategies to reduce this number. The use of Learning Objects (LO) has gained ground in higher education, but few experiments have been developed to help teach programming concepts to students in the early years. In other experiments, we applied the use of LO in the teaching of programming concepts. However, one of the difficulties encountered in our previous experiences was to identify the ideal moment to introduce the LO to students. Therefore, in 2015 we conducted a new experiment with two groups of students, using a LO to teach Sorting concepts. For the first group we introduced the LO before the professor started teaching theoretical concepts. For the second group, the professor initially taught the theoretical concepts, and only later was the LO introduced. This study aims to present the methodology used to carry out the experiment and the results achieved.**

Keywords – computer programming; computer science education; educational process; software algorithms; learning objects.

I. INTRODUCTION

Introducing algorithms and programming concepts to first year students has become a critical objective to Engineering and Computer Science courses. It is difficult for students to understand some concepts they are taught for the first time, such as logical thinking, abstraction, algorithms, data structures, formal computer language, and others. This is potentially quite challenging material that is going to form the basis of the rest of their learning. In few weeks students are introduced to data structures, programming resources, binary trees, sorting, which are examples of very important subjects for them, but many students do not learn those concepts appropriately.

The primary target of initial programming courses is algorithm design. As these are considered core courses for the Computer Science undergraduate, it is extremely important that students can clearly understand all the concepts covered. At our institution, students complete the first two years of programming concepts using C++ and they are introduced to all the knowledge about procedural and object-oriented programming and additional data structures and algorithms.

New Computer Science students usually have difficulties understanding and abstracting the problem logics and transforming it into a step-by-step sequence in order to develop the algorithm later. Several students have indicated that learning algorithms is the most difficult and the least interesting subject during their first year in Computer Science course [1].

The paradigm created inside and outside classrooms makes students come to first classes with the fixed opinion that this course will be a great obstacle, extremely difficult to overcome. Besides, some teaching methods used by teachers to present the concepts require great abstraction ability from the students [2].

Several reasons for student failures in learning algorithms and programming concepts have been described in the literature [3], [4], and most of them are related to undisciplined study habits focused on memorizing, unstructured previous knowledge (especially on the domains of mathematics and logics), non-motivating teaching approaches, difficulties in understanding problem wording, and high levels of abstraction.

The misunderstanding of programing concepts lead a significant percentage of students to drop out from their courses [1]. At our institution, the dropout rate in the first year of the Computer Science course is 26%, which led us to think of pedagogical strategies to reduce this number.

During the last years, we have conducted several studies about Learning Objects (LO) to help teaching computer programming concepts to students in early grades of Computer Science course [5], [6]. In this sense, we have developed LO to support teaching Pointers, Data Structures, Binary Trees and Data Classification.

However, one of the difficulties encountered in our previous experiences was to identify the ideal moment to introduce the LO to students, so that they could be more successful in using it. Therefore, in 2015 we conducted a new experiment with two groups of students, using a LO to teach Sorting concepts. For the first group, consisting of 27 students, we introduced the LO before the professor started teaching theoretical concepts. For the second group, made up of 31 students, the professor initially taught the theoretical concepts, and only later was the LO introduced.

This paper presents our experience in carrying out the two case studies, the results obtained with the development and implementation of a learning object in the teaching of Sorting concepts in a Computer Science course, as well as the differences between the results achieved with the application of LO before teaching such concepts and after it.

This paper is organized as follows: Section II presents the background and related works; Section III presents the Bloom's Taxonomy; Section IV presents the methodology, which includes the experimental procedures and the subjects; Section V and VI presents the data and results; Section VII discusses the results; and finally, Section VIII presents the conclusions.

II. Background and Related Works

Many efforts have been made to explore alternative methods of teaching programming concepts to students of the initial years. Teachers have tried the adoption of new tools and pedagogical approaches to help students engage in a more pleasant learning process. In the last years, the use of LO has gained a lot of attention as a common format for sharing educational content.

A LO element is a pedagogical resource, a type of computer-based instruction grounded in the object-oriented paradigm, usually delivered over the internet, meaning that any number of people can access and use it simultaneously. A LO is any digital resource that can be delivered across the network on demand and that can be reused to support learning [7].

To facilitate the widespread adoption of the learning objects approach, the Learning Technology Standards Committee (LTSC) of the Institute of Electrical and Electronics Engineers (IEEE) provided this definition: "Learning Objects are defined as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. Examples of technology-supported learning include computer-based training systems, interactive learning environments, intelligent computer-aided instruction systems, distance learning systems, and collaborative learning environments. Examples of Learning Objects include multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning" [8].

A LO can be used in a considerate manner, addressing a part of content that will be worked with students. The instructor can use it as an additional way for teaching.

Information technology and communication have contributed to various areas of knowledge; in particular, the field of education stands out. With regard to teaching basic concepts of computing, purpose of this study, the literature contains some efforts that aim to assist in the teaching and learning processes: Scratch is an environment that provides the construction of programs and was created with the aim of facilitating learning to beginners in programming. It was developed at MIT (Massachusetts Institute of Technology) using the programming language Squeak; Greenfoot, developed by the University of Kent, in England, and the University of Deakin, in Australia, aims to teach object-oriented programming from the construction of scenarios using the prepared environment for game development.

Greenfoot assumes that the student has previous knowledge of basic programming concepts [5]; the University of Wisconsin-Madison has developed Gamestar Mechanic, an online platform to build games. The focus is on fostering problem-solving skills and creating motivation for learning science, technology, engineering and mathematics; and finally, ALICE is a 3D programming environment developed specifically for students who have their first experience with object-oriented programming. The software allows the student to learn fundamental programming concepts creating animations and games. In ALICE, 3D objects such as people, animals, vehicles etc. form a virtual world where students create programs to animate such objects.

Experiments with the use of these software point to the fact that, despite the success of such experiments, much remains to be done in this direction [6].

An alternative that has contributed to the teaching-learning process is the use of LO. These features tend to facilitate and provide technological support to educational processes.

Several LO have been developed for various areas of knowledge, such as mathematics, biology, physics, chemistry, health, etc. There are many repositories for learning objects available worldwide, like DOOR, DSPACE, Fedora, EPrints, Edutools, Ariadne, and others. Most of them have been developed by the initiative of universities and governmental incentives, for example the BBC Learning, BozemanScience, Harvard Open Learning Initiative, Khan Academy, Learn NC, Merlot, Math Open Reference, MIT Open Courseware, Nobel Prize Education, Smithsonian Education, among others. A repository of LO is an organized collection of digital documents, to which the entire community may or may not have free and open access, and it can be seen as a digital library. Students can access these resources made available by the teachers, in order to assist in the course teaching-learning process.

But there is a very small number of LO developed to teach the fundamental concepts of computer programming, further enhancing the learning difficulties of students in the early years of Engineering and Computer Science courses.

In the last years the authors of this work have been dedicating to the development of LO to support teaching computer programming concepts to students in early years of Computer Science programs [5], [6]. We have developed and applied LO oriented to supporting the teaching of computer programming concepts. However, one of the difficulties encountered in our previous experiences was to identify the ideal moment to introduce the LO to students, so that they could be more successful in using it, and this is the main reason for developing this work.

III. BLOOM'S TAXONOMY

Although there are several models that address the learning process, we chose Bloom's taxonomy as a tool that can help us control the learning objectives.

Bloom and his fellow researchers found that, at the same teaching conditions, all students learn, however, they differ in depth of learning and in the abstraction of the knowledge acquired. In accordance to [9], the reason would be linked to the strategies and organization of the learning process adopted to stimulate the cognitive development.

Bloom's Taxonomy identified three domains of educational activities: cognitive, affective and psychomotor. The cognitive domain involves knowledge and the development of intellectual skills. In this domain the individual acquires new knowledge, promotes intellectual development and grows new skills and attitudes.

The affective domain includes the manner in which people deal with things emotionally, such as feelings, values, appreciation, enthusiasms, motivations, and attitudes.

The psychomotor domain includes physical movement, coordination, and use of the motor-skill areas. Development of these skills requires practice and is measured in terms of speed, precision, distance, procedures, or techniques in execution. Thus, psychomotor skills rage from manual tasks, such as digging a ditch or washing a car, to more complex tasks, such as operating a complex piece of machinery or dancing.

Although the three areas have contributed to the work of educators to define the processes of teaching and learning, the cognitive domain is the most known and used.

During the 1990's, a new study revised and updated Bloom's taxonomy [10], hoping to add relevance for 21st century students and teachers. The categories that make up the structure of cognitive domain proposed by Bloom, are illustrated in Figure 1 and represent the expected learning results. It is a cumulative hierarchical framework consisting of six categories, each requiring achievement of the prior skill or ability before the next one, which is more complex. The categories represent a relationship of dependency between the mental processes levels, starting from the lowest performance level (Remembering) to the most elaborate (Creating), with substantial degree of complexity.

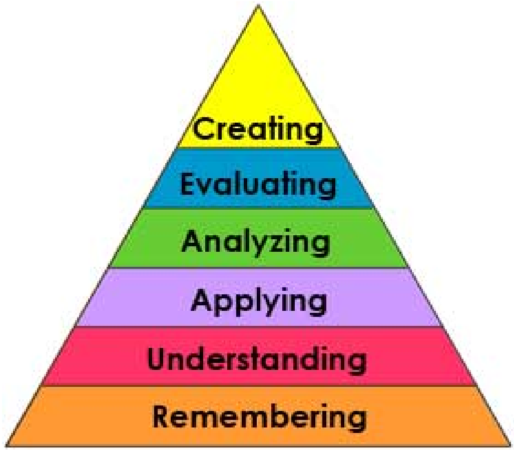

Figure 1 - Categories of cognitive domain of Bloom's Taxonomy

Below we briefly present the definitions of the elements that make up the structure of the cognitive domain proposed by Bloom and it is important for the development of this work:

1. Remembering: is the lowest level of learning in the cognitive domain and typically does not bring about a change in behavior. It involves memorization and recall of information with no evidence of understanding, but it is the building block of all subsequent levels of learning because the learner must remember information presented before progressing to the next levels. The activities for learning algorithms in this category can be bookmarking, flash cards, rote learning based on repetition and reading.

2. Understanding: learners comprehend the meaning of the material presented and predict consequences or effects from it. Although no change in behavior occurs at this level, learners are able to interpret prior learning, describe their understanding of what is presented and discuss how the new material learned may or may not work in their own environment. They can describe a concept in their own words, give an example or outline the steps necessary for some procedure. The activities for learning algorithms in this category can be participating in cooperative learning, taking notes, storytelling and internet searching.

3. Applying: learners effectively apply concepts, principles, methods and other newly learned information to novel and concrete situations in the form of measurable activity with minimal direction. In this stage, learners have the ability to demonstrate the correct usage of a method or procedure and a change in behavior occurs. Learning outcomes require a higher level of understanding than those in the Remembering and Understanding domains. The activities for learning algorithms in this category can be collaborative learning, creating a process, blogging and practicing.

4. Analyzing: learners have the ability to examine content and subdivide it into smaller pieces aiming to understand the final structure. They can separate material or concepts into component parts so that its organizational structure may be understood, and distinguished between facts and inferences.

The activities for learning algorithms in this category can be debating, questioning what happened and running a test.

5. Evaluating: learners have the ability to combine pieces of information not organized to draw up a "whole", they can make judgments about the value of ideas or materials, selecting the most effective solution. A big change in behavior occurs and learners can learn algorithms by themselves. The activities for learning algorithms in this category can be surveying and blogging.

6. Creating: learners can build a structure or pattern from diverse elements, put parts together to form a whole, with emphasis on creating a new meaning or structure. They have the ability to develop a new product or approach, making qualitative and quantitative evaluations of specific knowledge. The activities for developing this category can be creating a new model, writing an essay and networking with others students and professionals.

IV. METHODOLOGY

Our institution enrolls approximately 2,000 students in 11 programs. STEM-related degree programs are the most traditional and we have around 330 students enrolled at Computer Science course, with a typical first year class of around 100 students. In the first year, the Department of Computer Science offers an introductory programming course: Algorithms I, which is the expected starting point for students majoring in our Department. It is a full-year typical introductory course with no previous programming experience required. In the second year we offer the Algorithms II, closing the fundamentals for logic and programming basics.

One of the difficulties encountered in our previous experiences was to identify the ideal moment to introduce the Learning Object to students. In some cases we used it during the Algorithms I course, in other cases we used it at the end of the course. Therefore, in 2015 we conducted a new experiment with two groups of students.

We developed a new learning object for supporting the teaching process of Sorting to the second-year students of the Algorithms II course. With the use of interactive features, the LO works on the theory of data classification, algorithms developing, instructions, data and results, and simulation of sorting methods.

The sorting methods we are using in the LO are Bubble Sort, Insertion Sort, Selection Sort, Counting Sort, Shell Sort and Quicksort. When using the LO, the student has information about conceptual aspects of each method, running time, details about the algorithm, instructions for understanding it, graphical simulation and other important information. The Sorting LO is available at http://www.fema.edu.br/oa/Classificacao/HTML/.

We divided students into two groups. For the first group, consisting of 27 students, we introduced the LO before the professor started teaching theoretical concepts. Moreover, for the second group, made up of 31 students, the professor initially taught the theoretical concepts, and only later was the LO introduced. These two case studies took a two-month period to be carried out and at the end we collected the data for our research.

In order to evaluate the learning results, we created a framework based on Bloom's Taxonomy, to verify if the student acquired a competency or performance capability. It was important to ensure that learning activities and evaluation tests developed and assessed the same type of performance and learning content.

For each group of students, we assessed their competency for each one of the sorting methods of the LO: Bubble Sort, Insertion Sort, Selection Sort, Counting Sort, Shell Sort and Quicksort. Our focus was to verify the functionality of the LO and to check how useful it was in assisting in the teaching-learning process of students. Table I illustrates the general evaluation items we used for each sorting method; it was the starting point for creating the next evaluation criteria.

Table I – General Evaluation Items

Sorting Method	Category of Cognitive Domain	Learning Objective	Learning Activity	Test
Bubble				
Insertion				
Selection				
Counting				
Shell				
Quicksort				

The categories of cognitive domain are: Remembering, Understanding, Applying, Analyzing, Evaluating and Creating. The learning objectives and learning activities are related to each category, and the tests evaluate the success degree students reached.

For the presentation of the results obtained in the evaluation process, we will describe two case studies, the first in which we introduced the LO before the professor started teaching theoretical concepts about sorting methods, and the second which was conducted with the introduction of the LO after the professor had taught the theoretical concepts about sorting methods.

V. CASE STUDY 1

In order to know better the effectiveness of the most ideal moment to introduce the Learning Object to students, we conducted this first case study with 27 students in the Algorithms II course. For this group, we introduced the LO before the professor started teaching theoretical concepts. So, students still didn't have prior knowledge about sorting methods, except for those students who had studied by themselves or had participated in other course before.

This study was divided into five steps: to quantify the students' knowledge about the sorting methods; the use of the LO; the assessment of concepts learned after the use of the LO; the LO environment assessment; and the evaluation of student's competency growth, in accordance to the cognitive domain of Bloom's taxonomy.

1st Step: Identification of Students' Knowledge about Sorting Methods

In the first step, we applied the pre-test in order to identify the knowledge of students about the sorting methods. The pre-test consisted of theoretical and practical questions about this subject. The performance of the students after completing the pre-test is shown in Fig. 1.

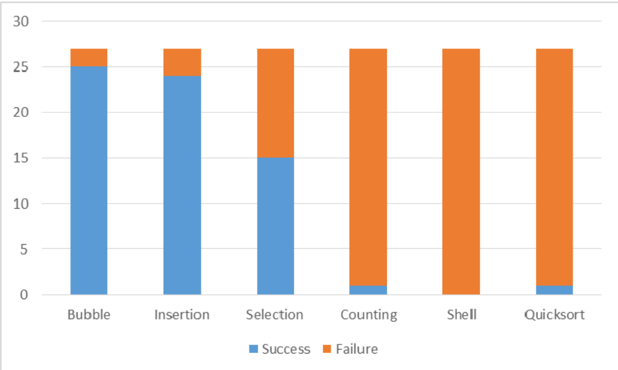

Fig. 1. Students' Knowledge about Sorting Methods – before the use of the LO

It can be observed from Fig. 1 that a higher percentage of students (92.6%, n=25) where successful when they answered questions about Bubble Sort, Insertion Sort (88.9%, n=24) and Selection Sort (55.6%, n=15), because they had been introduced to those concepts before. But very few students had some knowledge about the other sorting methods: Counting (3.7%, n=1), Shell (0.0%) and Quicksort (3.7%, n=1), as they had never been introduced to those concepts. Next began the activities regarding the second part of the case study.

2nd Step: The Use of the Learning Object

In this second step we conducted the application of the LO with the target audience. For the development of the LO, we started from a dialogical conception of learning, aiming to develop the skills and abilities of each student. So that students were able to set limits and capabilities for the application of the practical concepts, we adopted the following criteria: (1) create clear understanding of aspects related to the sorting process; (2) distribute them between everyday and scientific knowledge; and (3) allow students to dialogue with the acquired knowledge.

3rd Step: The Assessment of Concepts Learned

The post-test applied in this third step aimed to measure the knowledge acquired after completion of activities in the second stage, that is, if the developed LO fulfills its support role and aids in the learning of contents. We cared so that the pre-test and the post-test environments were similar in such a way that the differences could be attributed to changes in attitudes and also in students' learning. As the pre-test, the post-test had 6 questions on contents presented in Step 2. Fig. 2 illustrates the performance of students after implementation of the post-test.

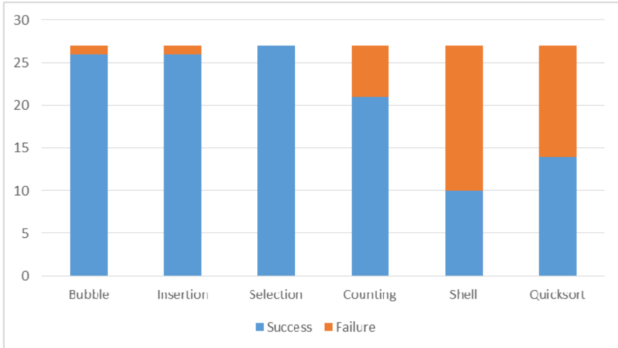

Fig. 2. Assessment of Concepts Learned – after the use of the LO

The post-test results show that after the use of the LO, we had a high percentage of success for all the students answering questions about the sorting methods. Even for Shell method the correct answers were 37.4% (n=10), we consider it was very positive, considering that students were introduced to those concepts for the first time. The post-test was important because it helped us in understanding which concepts and skills were well taught during the 2nd stage and which still require extra time to be worked on.

4th Step: Assessment of the Learning Object environment

In Step 4 students answered a questionnaire of satisfaction about the contents taught. They could assess how the LO was helpful in their learning, if it really was a feature that offered the support they expected, and how the experience in using the LO was. We developed the satisfaction questionnaire according to Likert scale, with five categories ordered and equally spaced. From the perspective of students, the survey results were encouraging, since 92% replied that the LO facilitates the understanding of the key concepts covered in the case study, and 96% of students see the LO as an important supplement to traditional teaching lessons format used by Computer Science course teachers.

5th Step: Evaluation of student's competency growth

Finally, in Step 5, we assessed the student's competency growth, in accordance to the cognitive domain of Bloom's taxonomy: Remembering, Understanding, Applying, Analyzing, Evaluating and Creating. Table II summarizes the results for the category "Understanding".

Table II – Competency Growth for category "Understanding"

Method	100%	80%	60%	40%	20%	0%
Bubble	74.1	22.2	3.7	0	0	0
Insertion	63	18.5	11.1	0	3.7	3.7
Selection	74.1	22.2	3.7	0	0	0
Counting	37	14.8	25.9	3.7	3.7	14.8
Shell	11.1	7.4	18.5	22.2	22.2	18.5
Quicksort	14.8	22.2	14.8	3.7	18.5	25.9

The results of step 5 show the understanding rate for each sorting method. For example, for the Bubble Sort, 74.1% of students understood it completely (100%), while that 22.2% understood it 80% and 3.7% understood it 60% after using the Learning Object. The same analogy can be done for the other methods. This information will be important for us to compare the results obtained in the second case study.

VI. Case Study 2

In order to know better the effectiveness of the most ideal moment to introduce the LO to students, we conducted this second case study with 31 students in the Algorithms II course. For this second group, the professor initially taught the theoretical concepts, and only later was the LO introduced.

The same strategy adopted on case study 1 was also adopted for this case. This study was divided into five steps: identifying the students' knowledge about the sorting methods; the use of the LO; the assessment of concepts learned after the use of the LO; the LO environment assessment; and the evaluation of student's competency growth, in accordance to the cognitive domain of Bloom's taxonomy.

It is important to notice that this second case study was conducted after the professor taught the theoretical concepts. So, students already had knowledge about the sorting methods.

1st Step: Identification of Students' Knowledge about Sorting Methods

In the first step, we applied the pre-test in order to identify the degree of confidence and knowledge of students about the content of Sorting Methods. The pre-test consisted of theoretical and practical questions about this subject. The performance of the students, after completing the pre-test, is presented in Fig. 3.

It can be observed from Fig. 3 that a higher percentage of students (64.5%, n=20) where successful when they answered questions about Bubble Sort, Insertion Sort (41.9%, n=13), Selection Sort (12.9%, n=4), Counting (32.2%, n=10), Shell (48.4%, n=15) and Quicksort (35.5%, n=11). Next began the activities regarding the second part of the case study.

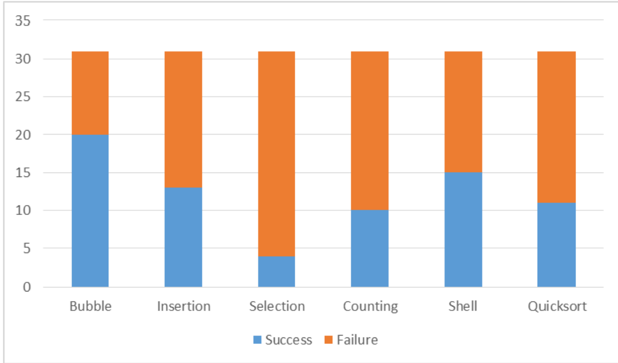

Fig. 3. Students' Knowledge about Sorting Methods – before the use of the LO

2nd Step: The Use of the Learning Object

In the second step we conducted the application of the LO with 31 students. This step was carried out in the computer lab and had the duration of two classes. For its implementation, theoretical and practical activities contained in the LO were planned.

It should be noted that the focus of the LO is essentially to present theoretical and practical aspects of Sorting Methods. In the end of each stage of the LO, students needed to solve some problems, using the concepts learned.

3rd Step: the Assessment of Concepts Learned

The post-test applied in this third step aimed to measure the knowledge acquired after completion of activities in the second stage, that is, whether the developed LO fulfills its support role and aids in the learning of contents. We cared so that the pre-test and the post-test environments were similar in such a way that the differences could be attributed to changes in attitudes and also in students' learning. The post-test had 6 questions on contents presented in Step 2. Fig. 4 illustrates the performance of students after implementation of the post-test.

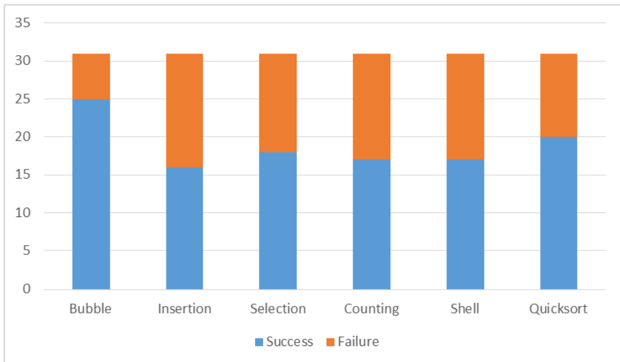

Fig. 4. Assessment of Concepts Learned – after the use of the LO

The post-test results show that after the use of the LO, we had a high percentage of success for all the students answering questions about the sorting methods.

4th Step: Assessment of the Learning Object environment

For this fourth step, we conduct an assessment of the entire LO environment. Students were asked to answer a questionnaire of satisfaction about the contents taught and assess how the LO was helpful in their learning, if it really was a feature that offered the support they expected, and how the experience in using the LO was. From the perspective of students, the survey results were very positive, since 97% replied that the LO facilitates the understanding of the key concepts covered in the case study, and also 97% of students answered that the LO is an important supplement to traditional teaching lessons format used by Computer Science course teachers.

5th Step: Evaluation of student's competency growth

In Step 5, we assessed the student's competency growth, in accordance to the cognitive domain of Bloom's taxonomy: Remembering, Understanding, Applying, Analyzing, Evaluating and Creating. Table III summarizes the results for the category "Understanding".

Table III – Competency Growth for category "Understanding"

Method	100%	80%	60%	40%	20%	0%
Bubble	80.6	19.4	0	0	0	0
Insertion	41.9	19.4	38.7	0	0	0
Selection	58.1	14	27.9	0	0	0
Counting	54.8	15.1	30.1	0	0	0
Shell	48.4	34.4	17.2	0	0	0
Quicksort	64.5	23.7	11.8	0	0	0

The results of step 5 show the understanding rate for each sorting method. For example, for the Bubble Sort, 80.6% of students understood it completely (100%), while that 19.2% understood it 80% after using the LO. The same analogy can be done for the other methods. This information will be important for us to compare the results obtained in the second case study with the first case study, and the analysis of this comparison will be presented in the next section.

VII. Results Analysis

On this section we will analyze the results obtained in the application of the two case studies. In the first one we introduced the LO before the professor start teaching theoretical concepts about sorting methods and the second one was conducted with the introduction of the LO after the professor had taught the theoretical concepts about sorting methods.

Table IV presents the results of the assessment of concepts learned by the students after the use of the Learning Object at both case studies.

Although we have assessed the student's competency growth for the six categories of the cognitive domain of Bloom's taxonomy, for this paper we have presented the results obtained for the category "Understanding", due to space limit.

The comparison between Table II and Table III takes us to Table IV, which summarizes the results obtained on Case Study #1 and Case Study #2.

Table IV – Comparison of results obtained on Case Study #1 and Case Study #2 for category "Understanding"

Sorting Method	LO Before theory Case Study 1	LO After theory Case Study 2	Growth
	60% to 100%	60% to 100%	
Bubble	100	100	0%
Insertion	92.6	100	7.9%
Selection	100	100	0%
Counting	77.7	100	28.7%
Shell	37	100	170.2%
Quicksort	51.8	100	92.9%

The values presented on Table IV gives us an adequate confidence level that the results obtained on Case Study #2 are more positive than the results obtained on Case Study #1, and represent a significant growth for the understanding of sorting methods. We can observe that the biggest growth index occurred for the Counting, Shell and Quicksort methods. They are exactly the same methods that students had more difficult to understand during the first case study.

VIII. Conclusions

This work aimed to present our experience in using learning objects for computer programming teaching in Computer Science introductory courses. One of the difficulties encountered in our previous experiences was to identify the ideal moment to introduce the LO to students. So, in 2015 we conducted this new experiment with two groups of students, using a LO to teach Sorting concepts. For the first group, consisting of 27 students, we introduced the LO before the professor started teaching theoretical concepts. And for the second group, made up of 31 students, the professor initially taught the theoretical concepts, and only later was the LO introduced.

These two case studies took a two-month period to be carried out and at the end, we collected the data to our research, whose results showed the ideal time for teachers to implement a LO.

For evaluating the learning outcome, we aligned the results of this research to Bloom's Taxonomy, to verify if the student acquired a competency or performance capability, which served as a basis for defining the objectives of the adopted curriculum and also the compliance of course evaluation purposes regarding local needs.

The results analyzed on Section VII are very clear and Table IV presents the results of the assessment of concepts understanding by the students after the use of the LO at both case studies, in accordance to Bloom's Taxonomy. We can conclude that the most appropriate time for teachers to implement a LO is after the theoretical concepts had been taught to the students. This was evidenced in the second case study.

The results presented on Table IV corroborate this conclusion, as we can observe the understanding growth obtained in the second case study for all the sorting methods. The methods considered more difficult by students – Counting, Shell and Quicksort, were those with the greatest understanding growth percentage when the students used the LO after the theory had been taught.

The results also showed that the use of LO can contribute significantly to the process of teaching-learning programming concepts to students in the early years, helping the understanding of difficult concepts and consequently assisting to reduce the failure and dropout rate.

The use of LO provides the creation of meaningful learning spaces. Meaningful learning is permanent and powerful, while learning detached from a context of meaning is easily forgotten and rarely applied in new learning situations or problem solving.

The results gathered on this project are very important and positive. They can serve as a basis for the academic community to start the development of more learning objects for teaching computer programming, making the learning process for first year's students less difficult.

As for future work, we think it is important to apply this study with more LO in different groups of students, so that we can verify if the results we obtained in the present study can be reached in other experiments.

REFERENCES

[1] Watson, C., Li, F. "Failure rates in introductory programming revisited". Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education ACM. 2014, pp. 39-44.

[2] Ribeiro, R. S., Brandao, L. O., Faria, T. V. M., Brandao, A. A. F. "Programming web-course analysis: how to introduce computer programming?". In 2014 IEEE Frontiers in Education Conference, October 2014, pp.1-8.

[3] Borges, Marcos A. F. "Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação." In Workshop de Educação em Computação – WEI 2000. 12 – 16 July 2000. Proceedings of the 20th Congresso Brasileiro da SBC. Vol. 1. Curitiba: Editora Universitária Champagnat, p. 64.

[4] Vihavainen, A., Airaksinen, J. and Watson, C. "A Systematic Review of Approaches for Teaching Introductory Programming and Their Influence on Success". Proceedings of the ACM 10th Annual Conference on International Computing Education Research, New York, NY, USA, 2014, pp. 19-26.

[5] Begosso, L. C., Begosso, L. R., Begosso, R. H., Ribeiro, A., Santos, R. M. "The use of Learning Objects for teaching Computer Programming". In: 2015 IEEE Frontiers in Education Conference, USA, 2015, pp. 786-791.

[6] Begosso, L. C., Begosso, L. R., Gonçalves, E. M. and Gonçalves, J. R. "An approach for teaching algorithms and computer programming using Greenfoot and Python". In 2012 IEEE Frontiers in Education Conference, October 2012, pp. 1-6.

[7] Wiley, D. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. Wiley (Ed.), The Instructional Use of Learning Objects. Bloomington: Association for Educational Communications and Technology, 2002.

[8] IEEE. Draft Standard for Learning Object Metadata. IEEE Standards Department, USA, 2002.

[9] Villanueva, I. "An exploration of Bloom's knowledge, skills, and affective-based goals in promoting development of freshmen engineering students' professional identities". In: 2015 IEEE Frontiers in Education Conference, USA, 2015, pp. 211-215.

[10] Krathwohl, D. R. A Revision of Bloom's Taxonomy: An Overview. Theory into Practice, vol. 41. no. 4, p. 212-264, 2002.