ACTA INFOLOGICA 2023;7(1):197-207
dergipark.org.tr/acin

DOI: 10.26650/acin.869786

RESEARCH ARTICLE

LISS'nAP: A Peer-Advising Collaborative Learning Approach for Object-Oriented Programming Learning

LISS'nAP: Nesne Yönelimli Programlama için Üç Aşamalı Akran Danışmanlığı İşbirliğine Dayalı Öğrenme Yaklaşımı

Rochdi Boudjehem[1], Ali Benyounes[2], Yacine Lafifi[3]

[1](Dr.) LabSTIC laboratory, University 8 Mai 1945, Guelma, Algeria
[2](MSc.) Computer Science Department, University 8 Mai 1945, Guelma, Algeria
[3](Prof. Dr.) Computer Science Department, LabSTIC laboratory, University 8 Mai 1945, Guelma, Algeria

ORCID: R.B. 0000-0002-9998-4833; A.B. 0000-0002-5821-1025; Y.L. 0000-0001-8232-4196

Corresponding author: Rochdi BOUDJEHEM
University May 8 1945 Guelma, Department of Computer Science
E-mail address: rochdiboudjehem@gmail.com, boudjehem.rochdi@univ-guelma.dz

Submitted: 28.01.2021
Revision Requested: 04.08.2021
Last Revision Received: 02.06.2023
Accepted: 02.06.2023
Published Online: 26.06.2023

Citation: Boudjehem, R., Benyounes, A., Lafifi, Y. (2023). LISS'nAP: a peer-advising collaborative learning approach for object-oriented programming learning. *Acta Infologica, 7*(1), 197-207. https://doi.org/10.26650/acin.869786

ABSTRACT

Among the existing programming paradigms, two of them are the most familiar and the most used by the majority of programers: procedural programming and object-oriented programming. To teach students programming, most existing curricula, especially academic ones, begin by teaching the basics of procedural programming, before moving on to introduce the concepts of object-oriented programming. Unfortunately, this transition is not always smooth for students, and the concepts of object-oriented programming can be perceived as problematic and confusing for some students who have trouble adjusting to the new programming mode and fail to find a decent object-oriented model for a given problem. Some scientists believe that the problem lies in the transition of vision when modeling problems from one vision that is familiar to students to a new one that is object-oriented. Some scientists think that the problem lies in the transition itself when modeling problems from a familiar vision to a new object-oriented one. Other scientists claim that the problem may not reside in the object-oriented paradigm itself, but rather in the existing tools available to teach it, such as the languages and the environments. In this article, we propose a new collaborative learning method for learning object-oriented programming, which can be also used to teach any other course that shares a similar pedagogical structure.

Keywords: Problem-based learning, collaborative learning, programming learning, e-learning

ÖZ

Mevcut programlama paradigmaları arasında, programcıların çoğu tarafından en tanıdık ve en çok kullanılan iki tanesi prosedürel programlama ile nesne yönelimli programlamadır. Öğrencilere programlamanın nasıl yapılacağını öğretmek için, özellikle akademik olan çoğu mevcut müfredatta, nesne yönelimli programlama kavramlarına geçmeden önce, prosedürel programlamanın temellerini öğreterek başlanır. Ne yazık ki, bu geçiş öğrenciler için her zaman "pürüzsüz" değildir. Nesne yönelimli programlama kavramları, yeni programlama moduna uyum sağlamakta güçlük çeken ve bir nesne için düzgün bir nesne yönelimli model bulamayan bazı öğrenciler için sorunlu ve kafa karıştırıcı olarak algılanabilir. Bazı bilim insanları, öğrencilerin problemleri bilinen bir vizyondan nesne yönelimli yeni bir vizyona modellerken, asıl sorunun vizyon geçişinde yattığına inanmakta. Diğer bazı bilim insanları ise, sorunun nesneye yönelik paradigmanın kendisinde değil, onu öğretmek için kullanılan diller ve ortamlar gibi araçlarda olabileceğini iddia etmekte. Bu makalede, nesne yönelimli programlamayı öğrenmek için yeni bir işbirliğine dayalı öğrenme yöntemi önermekteyiz. Bu yöntem, benzer pedagojik yapıya sahip başka derslerde de kullanılabilir.

Anahtar Kelimeler: Probleme dayalı öğrenme, işbirlikçi öğrenme, programlama öğrenimi, e-öğrenme

1. INTRODUCTION

E-Learning has been widely used in several fields and scientific disciplines recently. It has proven its effectiveness in making the learning process more flexible and manageable, especially during difficult times and emergencies that the world is witnessing more than ever, such as during wars or the spread of epidemics such as the COVID-19 pandemic. As in many fields, education has undergone substantial transformations, especially with the spread of Information and Communication Technologies (ICT). These changes can be seen through the colossal breakthrough achieved in the way the remote education field is presented, where the content of the course is increasingly adapted to the needs of each distinct learner.

Furthermore, specific characteristics of each taught subject are taken into account. Computer Science was among the first disciplines involved in e-learning, considering the ability of its learners to assimilate the use of this new learning form comfortably. Several e-learning environments and systems have been developed to cover different computer science topics, from logic and algorithmics to programming, including its different paradigms, especially the Object-Oriented Programming (OOP) paradigm. Many students often see OOP concepts as problematic, as they fail to develop good object-oriented designs for a given problem (Dillenbourg, 1999; Henri & Lundgren-Cayrol, 1998, 2001; Laal & Ghodsi, 2012; Rogalski, 1998). One commonly used technique to teach programming languages is Problem-Based Learning (PBL), which has proven effective in many educative fields like medicine and computer science (Koh et al., 2008; Walldén & Mäkinen, 2014; Wood, 2008).

Furthermore, it is always tricky for students to understand object-oriented concepts, as it is equally difficult for teachers to transmit them to students. Indeed, the difficulty is mainly related to the transition from the usual procedural modeling of problems to an object-oriented one. Research, as stated in the work of Stroustrup (1994), indicates that the programmer takes about an average of 6 to 18 months to change his mindset from a procedural to an object-oriented view of the world, which implies that the change is difficult.

For Kölling (1999), it is not object-oriented learning that causes problems, but the tools (languages and environments) available to teach it. For this author, the programming languages used are too complex, the programming environments (if they exist) are too confusing, and some of the systems and languages used for teaching were developed for professionals, which makes the situation more complex for students, who fall into the difficulty of using them in learning object-oriented concepts.

In the same sense, other authors see that for beginning programmers, as is the case for university students, identifying objects and creating an object-oriented program design is very difficult (Robins et al., 2003).

Unfortunately, most developed systems focus on the learner as an individual and have no interaction with his peers, making the learner feel lonely and increasing his sense of isolation. Learning with peers can positively influence the learner, which is well known in traditional or distance learning environments where many other skills can be developed, such as soft skills (Tadjer et al., 2020). That is why many works have been conducted on the benefits of using collaboration between learners inside distance learning environments, even though this kind of learning requires more effort and time than individual learning (Bremgartner & de Magalhaes Netto, 2012; Hank & Chikh, 2013; Hu & Xu, 2013; Lafifi et al., 2010; Troussas et al., 2014).

From a constructivist viewpoint, "Peer Interaction" or "Peer Dialogue" is a critical part of the "Peer Advising" course. Vygotsky's (1978) "Zone of Proximal Development" (ZPD) is one of the theoretical bases for "Peer Interaction." It states that social interaction between individuals helps them build their cognitive development, where individuals improve their skills through guidance provided by more experienced ones. Based on this assumption, "Peer Interaction" can provide a platform for peers to expand their knowledge by working together in socially constructed activities, as it allows peers to be "engaged in solving problems and building knowledge" (Merrill Swain & Lapkin, 2000). According to Villamil & De Guerrero (1996), peer-to-peer activities are the social pillar of cognitive process development. Indeed, the exchange of ideas during interaction allows peers to offer and receive help, which allows them to improve their knowledge. Furthermore, because dialogue occurs as part of a socially constructed knowledge-building activity "directly indicative of mental processes" (Merill. Swain, 1995), it becomes a primary source to understand how peers construct knowledge by rigorously analyzing the involved dialogue.

Peer advising involves “students helping students.” It highlights the “reciprocal interaction” of peers helping each other through collaboration during the learning process, taking a less formal, less-structured, and non-authoritarian approach. A peer advisor is a learner with more experience in a particular subject area who assists a peer student or a group of peers (less experienced learners) by guiding learning in that area. Although advisors and advisees learn together in the same course, the advisor assumes greater responsibility for providing guidance based on the needs of the advisee. According to Mynard and Carson (2014), there are at least three potential advantages that could be derived from peer advising:

1. Peer responses could be friendlier and more supportive than the teacher’s (Rollinson, 2005), making the learning atmosphere anxiety-free and comfortable and helping learners concentrate on learning (Ellis, 1994).
2. By embracing collaborative learning in peer advising, the advisor and the advisee could learn from each other in groups and work together to reach a higher level of performance (Donato, 1994). Furthermore, peer advising helps peers engage in a healthy relationship (Barman & Benson, 1981), where the advisor and the advisee share ideas and develop their knowledge together (Villamil & De Guerrero, 1996).
3. The Reciprocal Interaction of Peer Advising could benefit both the advisor and the advisee in developing their autonomy (Assinder, 1991).

This paper presents LISS’nAP (Listen-Up), a new approach that relies on Peer Advising and collaboration between learners to learn OOP concepts –or any other concepts with a similar pedagogical structure. It addresses these research questions:

- Is there an effective way to group learners to benefit more significantly from the learning system?
- If so, how many learners to assign to each group, and how to assign and assess them?
- Finally, How can a collaborative learning strategy improve OOP learning?

With those questions in mind, we propose to define a new approach that allows learners to deal with many difficulties experienced when learning this paradigm by helping each other. We propose a remote learning system that embodies and helps test the proposed approach.

This paper is organized as follows: Section 2 presents a literature review on programming languages and collaborative learning. In section 3, we give our proposed approach, and in section 4, we offer our conclusion and future works.

2. MATERIALS AND METHODS:

2.1. Online Object-oriented Programming learning

Object-Oriented Programming (OOP) is a fascinating and attractive yet crucial course for the curriculum of a university’s computer science students. It is one of the most popular programming paradigms and is highly demanded by technological development giants. However, learning the principles and concepts of this programming paradigm has proven to be tricky and sometimes complicated, leading researchers in this field to propose new approaches and methods to improve the quality of teaching this paradigm. In this sense, several approaches and techniques have been implemented in computer systems designed to solve the challenges and difficulties encountered when learning OOP. In Table. 1, we gathered and compared all found OOP’s existing learning systems.

After analyzing the collected works, we have concluded that most of the reviewed papers focused on individual learning techniques that treat each learner separately and omit the fact that the human being is a social creature who can learn by observing his peers and can acquire knowledge from the experiences of others. As for the “Peer Advising” perspective, most of the works we found in the literature used this technique in teaching human languages, and we found no work that used it to teach programming language concepts or paradigms. Indeed, these reasons motivated us to use “Peer Advising” in a new collaborative approach to help students learn OOP programming concepts.

Table 1
Comparative study of OOP's existing learning systems

Authors	Taught Course	Studied Programming Language	Used Technique	Implemented System Type	Implemented System Name	Online / Offline	Assessment?
(Cypher & Halbert, 1993)	*Procedural Programming*	/	*Visual & Example Based*	*Environment*	Pygmalion	*Offline*	*No*
(Benabbou & Hanoune, 2006)	*Algorithmic*	/	*Example-Based & Recommendation*	*Environment*	EasyAlgo	*Offline*	*No*
(Guibert et al., 2006)	*Procedural programming*	/	*Example Based*	*Environment*	MELBA	*Offline*	*Yes*
(Sims & Bubinski, 2011)	*General programming*	*Python, PHP, CSS, JavaScript, HTML*	*Online Learning*	*Platform*	CodeAcademy	*Online*	*No*
(Rose, 2003)	*OOP*	*Java, C++, Visual Basic*	*Visualization & UML Graphic Modeling*	*Tool*	Rational Rose	*Offline*	*No*
(Kölling, 1999a)	*OOP*	*Java*	*Visualization & UML Graphic Modeling*	*Environment*	BlueJ	*Offline*	*No*
(Esteves & Mendes, 2004)	*OOP*	*Java*	*Visualization & Simulation*	*Environment*	OOP-Anim	*Offline*	*No*
(Cheung, 2006)	*OOP*	*Java*	*Visualization & UML Graphic Modeling*	*Environment*	WEBLOOP	*Online*	*No*
(Yan, 2009)	*OOP*	*Java*	*Serious Game*	*Environment*	Greenfoot	*Offline*	*No*
(Djelil et al., 2015)	*OOP*	*C++*	*Serious Game*	*Virtual 3D Game*	PrOgO	*Offline*	*Yes*
(Seng et al., 2018)	*OOP*	/	*Serious Game*	*Mobile 2D Game*	Odyssey of Phoenix OOP	*Offline*	*Yes*

2.2. Participants and Data Collection

This paper proposes a new approach based on collaborative learning to assist students while learning the OOP concepts. In addition to assisting students in understanding and practicing OOP, it can help them overcome difficulties while learning this paradigm.

To reach the desired goals, we first conducted a survey study to uncover everyday problems and identify difficulties faced by students while learning the OOP concepts. Then, we proposed a new approach to assist students in overcoming difficulties while learning OOP using the survey results. After that, we implemented a new prototype system. Finally, we experimented with a group of students using the developed prototype.

The survey conducted with Computer Science students regarding OOP learning at Guelma University (Algeria) comprised 91 participants who answered 28 questions. The survey questions covered different topics about their experience with OOP learning to gather the required data. It was made available to students using two methods: online via a Google form and print-outs handed to students manually. The survey was prepared according to the Likert Scale guidelines for most questions (23 out of 28), using precise and well-crafted options to obtain the most reliable answers possible. The remaining five questions were either multiple-choice or with free answers.

3. RESULTS

3.1. Survey reliability and validity:

To measure the validity of the Survey, we used Pearson's Correlation Coefficient with N=48. In Pearson's Critical Table, with a Degree of Freedom- = N-2 = 46 and α=0.05, the critical value was 0.3044. Applying SPSS Bivariate Correlation Analysis, the obtained correlation table shows that 16 questions out of 23 were valid with Pearson's value > 0.3044 and a p-value<0.05 (Table. 2).

To measure the reliability of the Survey, we used the Cronbach's Alpha test, which gave us a Cronbach's Alpha value of 0.525, which is close to 0.6, indicating that the questionnaire is reliable (Table. 3).

Table 2
Case Processing Summary

		N	%
Cases	**Valid**	48	100.0
	Excluded[a]	0	.0
	Total	48	100.0

Table 3
Reliability Statistics

Cronbach's Alpha	**Cronbach's Alpha Based on Standardized Items**	**N of Items**
.525	.528	23

3.2. Analysis of the Survey Results and the Proposed Solution

We first analyzed the survey answers to identify the problems encountered faced by the students. After that, we used the survey results to propose a new approach to meet the research objective. Fig. 1 shows gathered statistics in some questions posed within the Survey.

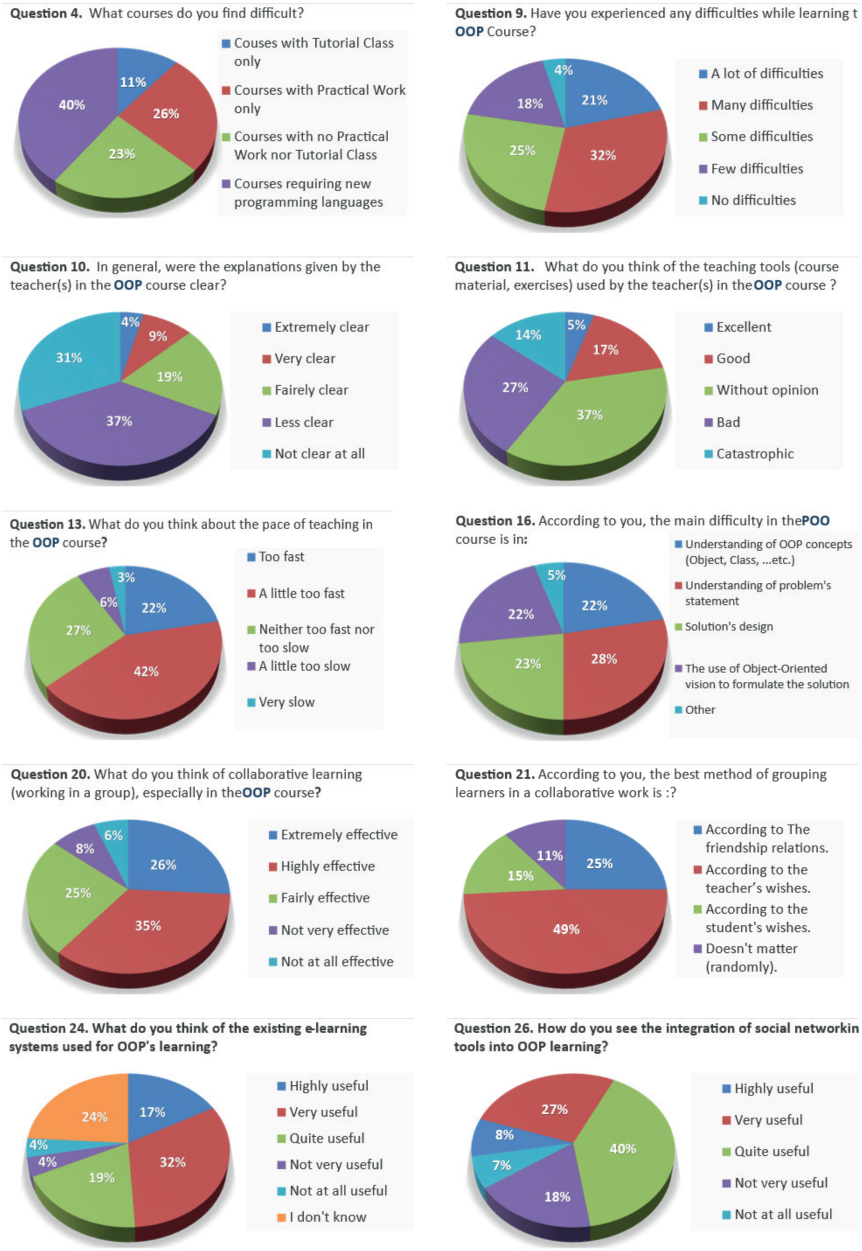

***Figure 1.** Samples of the statistics gathered from the Survey's Answers*

By analyzing the obtained answers, we got the following facts:

- Most students have difficulties mastering courses that require new paradigms and programming languages.
- In particular, most students have difficulties studying the Object-Oriented Programming (OOP) paradigm.
- 68% of the students did not understand the explanations given by the teachers in the OOP course.
- 41% of the students believe there is a problem selecting teaching tools for the OOP course.
- Regarding the pace of teaching in the OOP course, 64% of the students confirm it is faster than their own pace.
- As for the main problem in learning OOP, 51% of the students see that the problem is not learning the basic concepts of OOP but understanding statements of the problems posed in this module and arriving at an object-oriented design for these problems.
- As for using a collaborative solution (teamwork), most students (61%) think it can be effective in the OOP module. In addition, 49% saw that teacher-determined grouping is the proper method of grouping learners.
- Concerning the existing OOP's e-learning platforms' effectiveness, 49% of the students think they are helpful.

3.3. Proposition of LISS'nAP Approach (Learn, Imaging, Solve, Share, and APprove)

The proposed approach uses Object-Oriented Analysis and Design (OOAD) to teach OOP instead of a specific object-oriented implementation built on a particular programming language. The proposed LISS'nAP approach is a collaborative approach that relies on peer advising. However, before we go further into explaining the approach, we have to define the course structure and the grouping method.

3.3.1. Course Pedagogical Structure

The course to be studied should have two main stages:

1. **The Learning Stage:** the basic notions of the course are taught. The course is organized in the form of Chapters made up of a set of Concepts.

2. **The practice and application stage:** the acquired knowledge during the first stage is practiced, and Final Projects are implemented (Fig. 2).

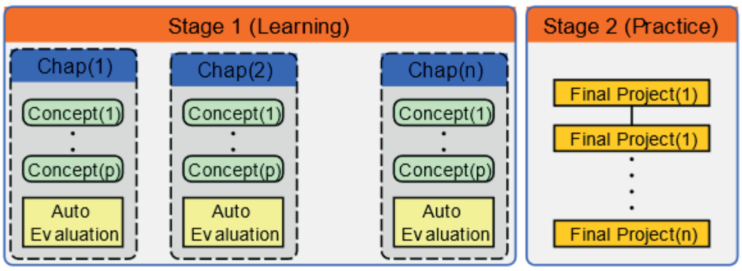

Figure 2. *Pedagogical Structure of the Course*

3.3.2. Group Formation method

When enrolling in the course, the learners have to answer a survey. The Survey is used to assess the learners' level of knowledge. Learners are then classified into five categories, "Excellent," "Good," "Average," "Fair," and "Poor." Learners' groups are formed of a maximum of 4 learners, and each group has to contain at least three different categories of learners. The grouping process takes place after the end of the two first stages.

3.3.3. LISS'nAP Sequence

Fig. 3 shows the two stages of LISS'nAP. Each of the four phases of the two stages is also shown in the figure. The two first phases: Learn and Imagine, are individual and do not require collaboration among learners. These two phases happen during the Learning stage of the course. The remaining phases: "Solve, Share 'n Approve," and Promote steps, are collaborative and occur in the second stage of practice and application, where the learners are perceived as a group.

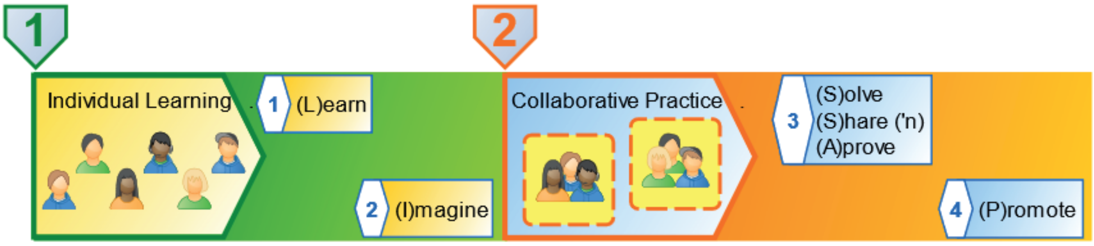

Figure 3. *LISS'nAP Sequence*

LISS'nAP is composed of a total of 6 phases that are the following (Fig. 4):

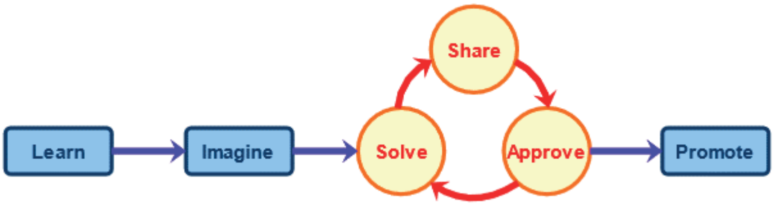

Figure 4. *LISS'nAP approach diagram with the Three Steps Cycle (Solve, Share 'n, Approve)*

1.a. (L)earn:

This first phase is individual. It is mandatory for the majority of educational courses. It is based on teaching the discipline's basic concepts, laws, and rules. It is part of the course's first pedagogical stage (Individual Learning Stage) to be studied, in our case, the OOP. The pedagogical content is organized into chapters and concepts (see Fig. 2). The learner begins to learn the basic concepts of OOP, such as classes, objects, and inheritance, using written content and grouped in well-prepared chapters maintained by the teacher. The latter, who plays the role of the course designer, must meet the following criteria: reliability and simplicity. He must also maintain a rate of progress and a rhythm of difficulty appropriate to the learning abilities of the learners. A self-assessment test quiz is handed out to the learners to evaluate their acquired knowledge at the end of each chapter.

1.b. (I)Imagine:

This phase is also individual and does not require any form of collaboration where each learner practices small tutorials with solutions to learn how to proceed with Analysis and Design step by step with minor solvable problems presenting a straightforward application of one of the concepts studied in the previous phase. Each learner is asked to use their imagination to extract the objects that constitute the solution to a given problem, determine the relationships between them, and then represent them in Unified Modeling Language (UML) diagrams.

Fig. 5 presents the four steps the learner follows to master the imagination.

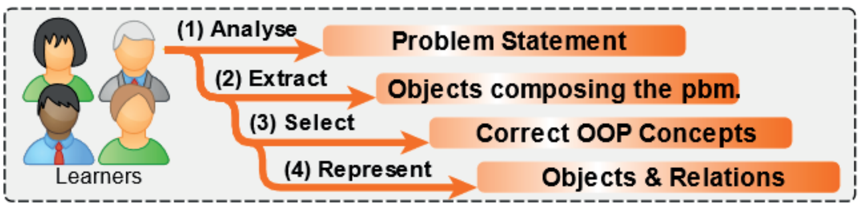

Figure 5. *Imagine Steps*

In this phase, the teacher prepares a set of solved problems using problem statements alongside their illustrative object-oriented designs detailing all the solution steps. The teacher explains how to extract the hidden objects in a problem statement, imagine their relationships, and correctly choose the right OOP concepts to present them.

Example:

The student can imagine the concepts and the relations among them from the following description: "*In a department, a teacher is responsible for a set of courses. The latter is composed of a collection of concepts and exercises. A group of students follows these courses".*

3. (S)olve, (S)hare (and) (A)pprove:

This phase is the first collaborative phase of our approach. This phase runs in a cycle for each learner, who is continually repeating the three-cycle peer advising steps: Solve, Share, and Approve, until he has the approval of his peers or he reaches the deadline of this phase. These three steps are as follows:

3.a. (S)olve:

In this phase, each group member tries to apply what they have learned in phase ***(I)magine*** by individually constructing object-oriented designs for the problem posed as a final project in progress. A real-time graphic design tool is made available to the learners, allowing them to understand better and simultaneously explain their designs to their peers in the next phase.

3.b. (S)hare:

After completing their designs, each learner shares his design with the other group members and awaits his peers' advice or approval. All the other group members have to visualize his solution and either approve his solution or advise him how to correct his design if this member judges the solution as erroneous. A third option is also available when the peer group member cannot give a solution so that he can abstain from voting. In that case, this member's vote is not accounted for, but he can always see the solution of his peer and can have an idea of the solution.

3.c. (A)pprove:

Each learner must advise other group members on their proposed solutions by approving or rejecting them while specifying the reasons behind the rejection. The solution is approved if the number of "approvals" exceeds the number of "rejections." Otherwise, the learner must repeat the cycle and continue improving his solution until getting "approved" or the exercise's time is over.

4. (P)romote :

After each group member finishes their specific design, it is time to choose the best design, promoted as the group's unique solution. Each member has to evaluate his co-members' solutions by ranking them using a star rating system. The voting process may follow these rules :

- The voting process begins after the solution is flagged as completed by the learner who owns the solution.
- The learners cannot rate their solutions.
- The voting operation must be completed before the deadline for the final project submission.
- If the project submission deadline is reached, and all the group members have not voted on any solution, the best solution is determined by a random draw between all the solutions marked as completed.
- The range of the score is from 1 to 5.
- The final solution score equals the sum of the voters' scores divided by the number of voters (the final score equals the average score).

- The solution with the best final (average) score is considered “the group’s solution.”
- In the case of equality between the average scores of two or more solutions, the best solution is determined by a draw.

Finally, the group solution will be sent to the lead teacher, who will assign a score regarded as “the group’s score.”

3.3.4. Collaborative and Individual Assessment

In addition to the collaborative assessment, which consists of a unique score for all the group members, another “individual score” is computed separately for each member, representing the learner’s engagement with his peers. It is measured using the frequency and the quality of his involvement in developing the collaborative solution. We must mention here that individual solutions, even though approved by the other members, may not have any value in individual and collaborative assessment. Doing so will discourage plagiarism or information hiding and promote the collaborative spirit against the “self-preservation” tendency.

3.4. Results of experimenting with the prototype:

We conducted a pilot experiment to validate and fine-tune the proposed approach. A set of four students at the Computer science department of Guelma University were evaluated before and after using the developed prototype using pre and post-tests (The experiment was conducted remotely during the first days of the Covid-19 containment period in Algeria).

The students had to take a pretest exam before using the developed prototype and another post-test exam after using the prototype. Table 4 presents the obtained results.

Table 4
Pilot Experiment results

	Student 1	Student 2	Student 3	Student 4
Pretest scores (/20)	**8**	**11**	**6**	**10**
Post-test scores (/20)	**12**	**15**	**9**	**13**

As presented in Table. 4, the post-test scores of the participants were better than their pretest scores, supporting that using the proposed approach helped them learn OOP concepts better, which was confirmed by the participants, who expressed their satisfaction with using this system.

4. DISCUSSION:

The main contribution of this study concerns the proposition of a new approach for enhancing the learning of Object-Oriented Programming (OPP) concepts. An online learning prototype adopted this approach. The pilot experiment with some students helped us enhance the system. Unfortunately, the proposed system was only made available to a limited number of students while they followed the OOP course online (during the Coronavirus confinement period, May-June 2020).

During this reduced experiment, we encountered many difficulties and faced many challenges with the students as they were challenging to understand and unpredictable. Some refused to work with others and declined any request for collaboration, while others who were too solicited felt exploited by their teammates, leading them to decrease their contribution or even stop collaborating. Also, the students’ grouping process proved tricky and presented a big problem because there were always cases where some students were highly requested while others were not.

The distribution of the tasks among the group members was also not very convenient and was often not even and not fair. Therefore, we sometimes found that the brilliant members did most of the work on behalf of the group while the weakest did not provide the required effort, which is not the intention of collaborative learning.

5. CONCLUSION AND FUTURE WORK

Our empirical study in the form of an online survey shed some light on some of the existing learning difficulties learners face when studying the Object-Oriented Programming paradigm among university students. Furthermore, this empirical study helped us design and implement a new method composed of two stages called LISS'nAP. The first one is individual, while the second is collaborative; each one includes a set of phases that enable the teaching of the basic concepts of Object-Oriented Programming and helps the learners practice what they have learned by encouraging them to solve real problems.

In the same perspective, and to implement our LISS'nAP method, we designed a system adopting this new method based on one of the Unified Modeling Language (UML) diagrams. The system in question is in the form of an e-learning system called LISS'nAP-POO. This platform provides learners with all the flexible and easy-to-use tools necessary to carry out the LISS'nAP method steps.

To check the effectiveness of the proposed method (LISS'nAP), we have scheduled a full-scale experimentation phase of the LISS'nAP-POO system with computer science students from a higher education institution. Furthermore, in future work, we propose to use artificial intelligence techniques for group formation. Using these techniques, we can get homogeneous groups meeting learners' expectations.

Peer-review: Externally peer-reviewed.
Author Contributions: Conception/Design of Study- R.B., A.B., Y.L.; Data Acquisition- R.B., A.B., Y.L.; Data Analysis/Interpretation- R.B., A.B., Y.L.; Drafting Manuscript- R.B., A.B., Y.L.; Critical Revision of Manuscript- R.B., A.B., Y.L.; Final Approval and Accountability- R.B., A.B., Y.L.; Material and Technical Support- R.B., A.B., Y.L.; Supervision- R.B., A.B., Y.L.
Conflict of Interest: The authors have no conflict of interest to declare.
Grant Support: The authors declared that this study has received no financial support.

REFERENCES

Assinder, W. (1991). Peer teaching, peer learning: One model. *ELT Journal*, *45*(3), 218–229. https://doi.org/10.1093/elt/45.3.218

Barman, C., & Benson, P. (1981). Peer Advising: A Working Model. *NACADA Journal*, *17*, 46–51. https://eric.ed.gov/?id=EJ253991

Benabbou, F., & Hanoune, M. (2006). EasyAlgo : Un environnement d'apprentissage et d'autoévaluation de l'algorithmique. In *admiroutes.asso.fr*. http://admiroutes.asso.fr/larevue/2006/76/Article_Apprentissage.pdf

Bremgartner, V., & de Magalhaes Netto, J. F. (2012). Improving collaborative learning by personalization in Virtual Learning Environments using agents and competency-based ontology. *2012 Frontiers in Education Conference Proceedings*, 1–6. https://doi.org/10.1109/FIE.2012.6462345

Cheung, R. (2006). A web-based learning environment for object-oriented programming. *International Journal of Information and Operations Management Education*, *1*(2), 140. https://doi.org/10.1504/ijiome.2006.009172

Cypher, A., & Halbert, D. C. (1993). What What I Do: Programming by Demonstration. *Cambridge, …*. http://regexp.bjoern.org/archives/000121.html

Dillenbourg, P. (1999). What do you mean by collaborative learning? chapter 1. *Collaborative-Learning : Cognitive and Computational Approaches*.

Djelil, F., Albouy-Kißi, B., Albouy-Kißi, A., Sanchez, E., & Lavest, J. M. (2015). Towards a 3D virtual game for learning object-oriented programming fundamentals and C++ language: Theoretical considerations and empirical results. *CSEDU 2015 - 7th International Conference on Computer Supported Education, Proceedings*, *2*, 289–294. https://doi.org/10.5220/0005475802890294

Donato, R. (1994). Collective Scaffolding in Second Language Learning. In P. lantolf James & A. Gabriela (Eds.), *Vygotskian Approaches to Second Language Research*. Ablex Publishing Corporation. https://books.google.com/books?hl=en&lr=&id=QbB-CGkx4hwC&oi=fnd&pg=PA33&dq=donato+1994+scaffolding&ots=FTssNtoMiN&sig=VZxAqXOq0jpPj8gcKWq_cSBRzBw

Ellis, R. (1994). *The study of second language acquisition*. Oxford University Press.

Esteves, M., & Mendes, A. J. (2004). A simulation tool to help learning of object oriented programming basics. *Proceedings - Frontiers in Education Conference, FIE*, *2*. https://doi.org/10.1109/fie.2004.1408649

Guibert, N., Girard, P., & Guittet, L. (2006). Performances et usages d'un environnement d'apprentisage de la programmation «basé sur exemple». *ErgoIA2006*. https://www.researchgate.net/publication/241141580

Hank, S., & Chikh, A. (2013). Towards a collaborative learning scenario adapted to the multiple learner's intelligence. *2013 3rd International Symposium ISKO-Maghreb*. https://doi.org/10.1109/ISKO-Maghreb.2013.6728135

Henri, F., & Lundgren-Cayrol, K. (1998). Apprentissage collaboratif et nouvelles technologies. In *Bureau des technologies d'apprentissage*.

Henri, F., & Lundgren-Cayrol, K. (2001). *Apprentissage collaboratif à distance*. PUQ.

Hu, L., & Xu, H. (2013). Collaborative M-learning based on multi-agent system. *Proceedings of 2013 IEEE International Conference on Service Operations and Logistics, and Informatics*, 59–62. https://doi.org/10.1109/SOLI.2013.6611382

Koh, G. C. H., Khoo, H. E., Wong, M. L., & Koh, D. (2008). The effects of problem-based learning during medical school on physician competency: A systematic review. *CMAJ*, *178*(1), 34–41. https://doi.org/10.1503/cmaj.070565

Kölling, M. (1999a). Teaching object orientation with the blue environment. *JOOP - Journal of Object-Oriented Programming*, *12*(2), 14–23. https://kar.kent.ac.uk/21734/

Kölling, M. (1999b). The problem of teaching object-oriented programming. *Engineering*, *11*(9), 6–12. https://kar.kent.ac.uk/21879/2/the_problem_of_teaching_object-oriented_kolling_1.pdf

Laal, M., & Ghodsi, S. M. (2012). Benefits of collaborative learning. *Procedia - Social and Behavioral Sciences*, *31*, 486–490. https://doi.org/10.1016/j.sbspro.2011.12.091

Lafifi, Y., Gouasmi, N., Halimi, K., Herkas, W., Salhi, N., & Ghodbani, A. (2010). Trace-based collaborative learning system. *Journal of Computing and Information Technology*, *18*(3), 207–219. https://doi.org/10.2498/cit.1001410

Mynard, J., & Carson, L. (2014). *Advising in language learning: Dialogue, tools and context*. Routledge. https://doi.org/10.4324/9781315833040

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *International Journal of Phytoremediation*, *21*(1), 137–172. https://doi.org/10.1076/csed.13.2.137.14200

Rogalski, J. (1998). Concepts et méthodes d'analyse des processus de coopération dans la gestion collective d'environnements dynamiques. *Communications Interactives Dans Les Groupes de Travail*, 27–58.

Rollinson, P. (2005). Using peer feedback in the ESL writing class. *ELT Journal*, *59*(1), 23–30. https://doi.org/10.1093/elt/cci003

Rose, I. R. (2003). *Visual modelling with Rational Rose*.

Seng, W. Y., Hayati, M., Yatim, M., & Hoe, T. W. (2018). Learning object-oriented programming paradigm via game-based learning game – pilot study. *The International Journal of Multimedia & Its Applications (IJMA)*, *10*(6). https://doi.org/10.5121/ijma.2018.10615

Sims, Z., & Bubinski, R. (2011). *Codecademy.(2011)*.

Stroustrup, B. (1994). *The design and evolution of C++*. Addison-Wesley.

Swain, Merill. (1995). Three functions of output in second language learning. In *Principle and practice in applied linguistics: Studies in honor of H.G. Widdowson* (pp. 125–144). Oxford University Press. https://ci.nii.ac.jp/naid/10025500767

Swain, Merrill, & Lapkin, S. (2000). Task-based second language learning: The uses of the first language. *Language Teaching Research*, *4*(3), 251–274. https://doi.org/10.1177/136216880000400304

Tadjer, H., Lafifi, Y., Seridi-Bouchelaghem, H., & Gülseçen, S. (2020). Improving soft skills based on students' traces in problem-based learning environments. *Interactive Learning Environments*. https://doi.org/10.1080/10494820.2020.1753215

Troussas, C., Virvou, M., & Alepis, E. (2014). Collaborative learning: Group interaction in an intelligent mobile-assisted multiple language learning system. *Informatics in Education*, *13*(2), 279–292. https://doi.org/10.15388/infedu.2014.08

Villamil, O. S., & De Guerrero, M. C. M. (1996). Peer Revision in the L2 Classroom: Social-Cognitive Activities, Mediating Strategies, and Aspects of Social Behavior. *Journal of Second Language Writing*, *5*(1), 51–75. https://doi.org/10.1016/S1060-3743(96)90015-6

Walldén, S., & Mäkinen, E. (2014). Educational data mining and problem-based learning. *Informatics in Education*, *13*(1), 141–156. http://www.scopus.com/inward/record.url?eid=2-s2.0-84903887180&partnerID=40&md5=81c98948c4908d535123ec2d7139cc0d

Wood, D. F. (2008, May 3). Problem based learning. *BMJ*, *336*(7651), 971. https://doi.org/10.1136/bmj.39546.716053.80

Yan, L. (2009). Teaching object-oriented programming with games. *2009 Sixth International Conference on Information Technology: New Generations*, 969–974.