https://doi.org/10.31449/inf.v43i1.2335 Informatica **43** (2019) 129–144 **129**

Collaborative Strategy for Teaching and Learning Object-Oriented Programming Course: A Case Study at Mostafa Stambouli Mascara University, Algeria

Chérifa Boudia
University Ahmed Ben Bella Oran 1
PB 1524, El M'nouer Oran, Algeria
E-mail: cherifa.boudia@univ-mascara.dz

Asmaa Bengueddach
University Ahmed Ben Bella Oran 1
PB 1524, El M'nouer Oran, Algeria
E-mail: asmaa.bengueddach@univ-oran1.dz

Hafid Haffaf
University Ahmed Ben Bella Oran 1
PB 1524, El M'nouer Oran, Algeria
E-mail: haffaf.hafid@univ-oran1.dz

Student paper

Keywords: university teaching strategies, oriented-object programming, collaborative learning,

Received: May 21, 2018

Several strategies and methods have been designed and tested to allow students to have better teaching and learning of programming concepts and training their programming skills acquired. In our university, old and classic pedagogical methods are still used in the learning of Oriented-Object Programming (OOP). This paper explores the experimentation of a pedagogical approach designed for Mostafa Stambouli Mascara University's computer sciences students to enhance their chances to get an effective teaching using a collaborative learning and early introduction of current technologies. The study was made of 108 second year informatics students' for two semesters (October–January 2016-2017, September-December 2017-2018) who were identified in a controlled experience for collaborative learning OOP in lab session. Students gathered into predetermined groups based on initial test and some setting. The students are supposed to share the same programming assignment by distributing roles according to global pedagogical scenario for problem-resolving activities. This experience allows the teachers to apply this strategy and see students in closer and permit them to express their problems and search together for solutions. The principal objective was to offer a new experience to motivate students and make this design a smart opportunity to have better programming performance. Furthermore, as a result of testing this new strategy, the students' academic progress is clear. A questionnaire was distributed with the purpose of analyzing students' motivation and satisfaction levels alongside the effects of the experiment. Results show that students found that among other benefits this method facilitates the learning process.

Povzetek: Poučevanje objektnega programiranja v Alžiriji na osnovi skupinskih pristopov.

1 Introduction

The challenges of the act of teaching are diverse and intricate in Algeria, as in most universities of the world [1], but remain focused on its pedagogical components [2], [3]. Indeed, to make our university credible innovative learning environment, newly recruited teachers are trained in teaching methodology for several months. However, these teaching methods and strategies are not deepened and stay premature [4]. Coming to computer sciences where problems are doubled. In fact, programming is a very hard task for students because it required high metacognitive skills like abstraction, deep comprehension, and long awareness, and many steps in problem-solving from analysis to debugging and testing [5]. In addition, teachers claim further difficulties found during programming lab sessions, due to several reasons such as the large number of students per class [6]. To overcome these challenges, collaborative learning is a good situation to involve students to help each other in groups for the same assignment activity by discussing ideas and giving everyone a simple task [5]. They can switch roles when the activity is completed. Consequently, Students enjoy the learning experience; feel motivated and ready for more improvement, in the other side, the teacher found more time to go further on

the course [7]. To elaborate collaborative learning in teaching programming for lab sessions, it is necessary to design a collaborative strategy for each step in problem-solving and hire metacognitive abilities respectively in every simple activity [8]. Based on this idea, the students' final program is the result of a collaborative strategy in teaching programming lab sessions starting from forming adequate groups which students work and learn from sharing, discussing their separated solutions and select the best solution that everyone has participated and understood. The overall purpose of the study is to demonstrate the impact and the effectiveness of a collaborative strategy on student learning in our university. As principal element, we introduce the collaborative learning in programming lab session course. As well, identify the behaviors of students through all designed roles in programming activities at different phases of global learning scenario.

The lack of experimental research of new educational strategies in the field regardless the poor level of students' academic performance in programming. The decision was made to experiment a new strategy and to reveal the results. Our perspectives, as teachers experience's in teaching programming, enable us to discover what challenges are presents, and the strategies claimed to be using successfully. In addition, a variety of pedagogical strategies was recommended from teaching practices. In categorizing approaches taken by teaching to support students, five key themes emerged globally collaborative learning, unplugged type activities, contextualization of tasks, developing computational thinking, and scaffolding programming tasks. Hence, this study aims to investigate the following research questions:

Will collaborative learning with different roles in lab design affect students' academic performance in programming?

Can resolution strategy process or steps guide the students in their achievement?

Is there a relationship between the uses of ICTs to gain computer metacognitive competence?

What is the teacher role in the entire experience?

The rest of the paper is organized as follows: section 2 describes the state of art of the research topic, related works and common definitions; section 4 displays the Algerian universities context about teaching computer-programming course. Section 5 discusses the methodology and its results of the strategy. The last section, presents some conclusions and perspectives associated to the result of our research.

2 Related work

2.1 Teaching OOP courses

In computer science, the teaching strategies and methods should be completely redesigned especially for programming courses [9]. Relatively to the nature of programming that presents challenges for both teachers and students [10]. It's resulting mainly by the gap between theory and practice [3] in converting the understanding OOP courses' lectures words into specification and coding using keyboards machines. The students seem to be disconnected from the practice of programming in lab sessions, where the smallest difficulties related to misunderstanding of a small semantic detail. Moreover, they are not diligent to work with current technologies in place of known tools [11]. Another difficulty for many programmers' students is the multiplicity of approaches, and tools for program development. An approach or an algorithm that works in one case will sometimes not work in another, even though the problems appear review similar. The identification and selection of alternative approaches is seldom. Most programmers must rely on experience to determine the appropriate way [12].

To facilitate an effective learning experience in programming, and to support students understand the concepts, gain skills and promote their thinking performance, we need a collection of components beside the pedagogical ones. Researchers argues the necessarily of ICT (Information, Communications & Technology) tools in building a solid environment for programming education [13], [15]. Some tools are really serving the whole education process for a good comprehension of concepts and developing programming skills such as game [14], [13] [16], [17], [18]. Education is still, at an important percentage, teacher-centered learning where he is the source of knowledge, key of participation, students' engagement and collaboration [19]. The use of internet and especially Facebook emerged significantly education to student-centred learning [20]. It was proven that providing links and materials is more useful for student to learn and give them extra time outside university doors for communication between teacher and students and among students [21].

Teaching guides the student' progress from beginner [10] to expert by attaining skills, and develop intelligence, and learn to communicate with teachers and his class members [22], [23], [24], explore his own conviction and attitude, and recognize his prospective, complete the task to another high one [25], [26]. As well, learning to be a computer programmer is considered a very hard task especially for beginners [27]. It requires constant improvement in programming skills. If learners are not moving forward, they are going backward.

Learning to program involved many techniques, like verifying arguments using methods provided by programming tools [10], understanding the important each code blocks, experimenting that is performed by executing some test data, learning more by reading the outputs of the program to find errors in the software code, and using the open source codes, in both books and sites that host and release millions of lines of software [5]. Some of those programming techniques can help in a specific pinch, while others focus on the programming tools and environments. Despite the nature of each technique, when used with awareness and diligence, each can help develop students abilities, both beginner, and expert [28]. In the same time, some students show resistance to develop the competencies required to code programs than execute it properly [17]. Consequently, it

is imperative to know what barriers make this kind of learning complicated and how students could learn correctly and easily [29]. Furthermore, program execution on a machine is a dynamic process and it is complex to evoke mentally all the treatment seize and track how variables change during program execution [9], [30]. Others difficulties like problems in visualizing all the changes that happen when the program is running. Programming engages problem-solving strategies [31], choosing appropriate resolution methods, coding, debugging and testing a program as a result [32], [15].

Besides, programming courses traditionally emphasize theoretical basis understanding of OO programming concepts, as well as its application in code fragment [33]. The concepts are truly learned during the practical experience to develop very high programming skills level, students needed to do often practice on programming exercises and to master debugging [1]. This course is presently brought to students in different strategies: traditional face-to-face learning [34], online learning [25], or in both situations called: blended learning [35].

2.2 Collaborative learning

Definitions

The collaborative learning is a situation where an interaction between students with exchanging, sharing ideas and information, distributing tasks [5] in order to achieve learning outcomes [37]. The importance of collaborative work is in summoning all the effort for the complementary relationship that conducts to excellent results. In collaborative learning situation, where at least two students work together to solve the same assignment, engage in a discussion in which all elements are working together to resolve a problem, and agreed on the founded solution at the end where various roles can be considered [28]. In psychological sciences and education, a study of Temperman and his colleagues [37] suggested six roles specific to learners according to their learning style (coach, organizer, secretary, moderator, administrator and theorist) in a collaborative remote learning environment. Andriessen [38] talk about discussion through visual representations that can help students learn to think critically and independently, even when students disagree, they still share the common concepts and they are all interested in achieving the same goals, in the meantime, the teacher should supervise the whole session and intervene depending on the problem to solve.

Collaborative learning is used in many education disciplines, rather in engineering science. Computer science courses, mainly in face-to-face lab sessions, when the students are all presents, roles can be distributed at the beginning of each problem-solving and switched at the end. Literature mentions different collaborative learning strategies in programming courses [30], [7]. The most used is pair programming, two students share the same computer and attempt to solve the programming exercise, one of them the ”driver” pilot the coding task and the other the “navigator” detect and correct the eventual errors [39]. When the group is made up of more than two members in small or big groups, different tasks could be proposed according to the collaborative learning activities scenarios or scripts [40]. The distribution of those tasks might be automated in traditional teaching environments[41]. Another technique is TPS (Think-Pair-Share). Students operate in three phases: think and work individually for few minutes on the task, then engage in discussion with neighbors to write the detailed code encouraged by the teacher. At the last phase, the entire class participates in discussion for the proposed and alternative solutions. It is simple and easy to deploy in large class for small groups used in lectures and labs [42]. Learning computer programming accrue many abilities such as thinking critically, analyzing and synthesize information, organizing and planning in groups. Those abilities depend on others skills like social skills, it enhances communication and motivation form group working [43]. A collaborative environment reinforced by ICT tools increase interaction between learners [17]. Learners can discuss the subject problems, share ideas and/or code fragments through online interaction [44], [27].

Collaborative programming on the same programming assignment activity consists on designing several pedagogical scenarios depending on group settings. Many settings for group formation exist: random selection, self-selection, homogenous, and heterogeneous. Random selection generally is used in first year programming courses curriculum or large classes [9], where the students meeting up in groups set from the list of registered students ordered alphabetically. The random segmentation for large classes provides equitable distribution easy to apply offers to students, the same chance to interact socially being in any group [45]. Self-selection is used frequently project work groups. The students are more comfortable in choosing their mate based on friendship, previous programming experience, and previous studies [28].

In the homogeneous group setting, members join a group where similar characteristics and preferences are in common such as learning style, same academic level [36], [23], [46]. Among its advantages, students can promote actively each other by interacting with varied individuals to enhance their abilities [39]. The other sides, students with different characteristics like learning level, make heterogeneous groups. This dissimilarity may be in each student but the same learning objective brings them together. The conditions that determine the choice of the group depend on the learning activity’s objectives [5], [47]. In literature, mixed groups are the most adopted setting [28]. However, the problem lies in heterogeneous elements' determination to configure the group in question [23]. Therefore, frequent researches have experienced the effect of group formation method on group performance [22], [48]. At this point, the importance of identifying witch settings may lead to a fruitful learning experience.

All the reviewed studies provide a big evidence of how academics and teachers are agreed to work hard in order to improve teaching and learning computer

programming. Though, they operate at different aspects and use different techniques on teaching methodologies. In such a complex field as education and pedagogy, we focus on two aspects: techniques and methods in teaching computer programming and collaborative learning. Some parameters are needed to determinate a clear vision of each study, such as scope, timeframe, etc. From this state of art, collaborative learning may be a good practice to involve in learning programming. In methodology section, we will provide all settings to define our pedagogical approach.

Pedagogical scenario

The pedagogical scenario is a central concept called method that defines the learning unit's scenario. This meta-model contains the descriptors necessary to model a learning unit where the role of each intervener is precisely described as well as the objectives and prerequisites of each participating activity to achieve a global goal [49] like in a collaborative learning situation. Therefore, the pedagogical scenario is the product of a design process. Its content consists of objectives, a planning of learning activities, a schedule, a description tasks to be carried out by the students, evaluation methods all defined, arranged and organized during a design process. The Scenario is considered as a structured and coherent set of two parts: the learning scenario and the coaching scenario. The learning scenario' role is to describe the activities designed for use by learners and their assembly to build a learning situation and the productions that are expected. The frame scenario, or scenario as called by [50], specifies how teachers intervene tutors as designed to support the learning scenario which is considered to be a specification guiding the progress of the activity in the IT environment for which it was designed [51].

The passage from the simple textual description describing the activity towards its unfolding through a series of transformations that plays the scenario in the environment selected target computer. In [52], the authors define the pedagogical scenario as the description of a learning situation in terms of roles, activities and environment necessary for its implementation, but also in terms of knowledge manipulated. Also, they make the difference between two types of scenarios: the predictive scenario: defined as established before by a designer for the implementation of a learning situation, and the descriptive scenario defined as a scenario that describes the progress of the situation a posterior learning by including in particular the actors' activity traces (mainly the learners) and their productions.

2.3 Algerian universities practice in teaching computer programming

To properly conduct our study, we need to make a diagnosis on our specific problem we are facing and carry out a state of existing to see what strategy of teaching that could success in our environment with our students with the available components. Since 2004, almost in all higher institutions and universities in Algeria, educational system had moved to LMD system (licence, Master, Doctorat). Many malfunctions has accompanied the introduction of this new system, experts and academics argues from an evaluation of this experience in each university, and proposed, in terms of actions and measures that will allow the university to fill in the gaps and overcome those problems [53], this paved different features as a reform at the national level. At Mostafa Stambouli Mascara University, the undergraduate curriculum in computer science has been completely redesigned in the past few years. As part of this redesign, in the first semester, students are taking initiation to algorithmic at the first-year mathematics and informatics. This course introduces basics concepts in computer science as algorithms and basics data structures that include one lecture, one tutorial and one lab session weekly. In second semester, they take, as continuation of initiation to algorithmic, programming and data structure. This course gives a deeper notions and data structures using C programming language. Those two courses are ones of several fundamental teaching units' courses. Among methodological teaching units, a new course (M211) Information and Communications Techniques where students can distinguish new technologies for information and communication. As elective courses, students can choose between (M212) programming tools for mathematics and (M213) Introduction to Object-Oriented programming only one lecture in a week. Because students don't take any elective course in the first-year, they took these courses mandatory, M213 in the second year informatics and M212 in the second year mathematics. They are required to take oriented-object programming course with Java as the programming language with overall hourly volume of 67 hours and 30 minutes. The lectures were optional to attend but tutorials and lab sessions were mandatory. At the beginning of the semester, for the second year informatics class, students are divided into groups by alphabetical order; the average number of students per group is 20 according to the number of students enrolled. The two machines rooms granted to the informatics' students are rented for all courses with lab session within one hour and thirty minutes period weekly. The machine rooms are equipped with computers, and their tables, a whiteboard and a desk for the teacher. This course is generally given by one teacher who is in charge of the lecture, tutorials and the lab sessions or no more than two teachers; one in charge of the lecture and the tutorials, the other in charge of the lab sessions. The official program of this course contains five chapters: introduction to OO Programming, classes, heritage and polymorphism, interface and implementation, and graphic interface and applet with java programming language. The student is expected to acquire during this course the following skills:

1- The fundamental of object programming in java,
2- Reading and understanding programs in java,
3- Writing a program in java as a solution for a given problem,
4- Writing sophisticated applications (use of advanced data structures).

The course's assessment is done in two forms Continued Control (CC) and the final exam. For tutorials and lab sessions respectively, the teacher must do two CCs that could be a tests in machine room or homework and provide the marks before the final exam.

3 Method

The objectives of the study are to demonstrate the impact and effectiveness of implementing the collaborative group in learning achievement. Characterizing the nature of student behavior through different roles in programming learning, whether the teaching techniques are useful for students in face-face and distance learning, and to carry out a comparative study at different phases of global learning scenario with teacher and students engagement involved in.

3.1 Groups' determination

In this study, 108 students of two different semesters (about 14 weeks per semester) from second year informatics undergraduate OO programming course with Java programming language are involved. The experience is conducted only on lab sessions according to a schedule time. Each semester for lab sessions, students were initially divided into 3 groups by alphabetical order. Separately, each group from the three initial groups, attend the lab session consecutively. As indicated in Table 1, Only 55 students (27 females, 28 males) from 67 enrolled students in semester one and 53 students (22 Females, 31 males) from 55 enrolled students in semester two. At the beginning, the students are randomly divided into 3 groups, and each group is composed of 22 as an average. Starting by setting up the sub-groups according to three steps:

The initial test (IT)

As the first step, it consists of making a simple MCQ that includes 16 basis algorithmic questions. It aims to determinate the performance Academic Level AL of each student. Another four questions, which are the age, score in first-year, color, and the last one, is: do you want to be the representative of your group? The color preference may identify similarities for the same group [54]. The answer of the last question can reveal the leadership potential of students [55]. As a result, the MCQ allows us to classify in 3 categories according to scores' intervals: good [16-11[, medium [11-6[, poor [6-0].

Semesters	Total enrolled students	Total remain students	Initial groups	Sub-groups
1	67	55	3	14
2	62	53	3	17

Table 1: Information on the number of students and groups formed students.

Information examination

At the stage, we have all information like AL according to IT. We make the students together by followings AL and colors preference in a way to formulate in each color group for example: blue group contain one good, two mediums and one poor. Remark: to have more balanced groups, we tried to mix the two types of gender in each group.

The final step

In class, we told the students witch group they belong to. If they do not have any other issue about their partners, any preference or suggestions, they join their groups, sit together, and share one computer. According to this, sub-groups are formed of 3 to 5 students each. The teacher recommended all groups to get textbook called Pedagogical Notebook (PN) in order to note every lab sessions activity, they cover it with their favorite color. The teacher checks frequently the PNs notes and gives extra marks for the best pedagogical notebooks at the end of the semester.

3.2 Roles distribution

We start to give details how to work in the group with explanation the roles of each member inter and intra group.

The students' roles

We have determinate a variety of roles that the students must know as programmers and as members in the group. Those roles are quite different from most studies, such as those reported by [37]. The roles are not assigned but taken after discussing. Students' roles are the main piece in our collaborative learning design because when all group members discuss and get a task, the student assumes his work with determination. Roles should rotate at the end of the exercise in order to allow every student can perform and experience all roles so that everyone takes a turn and consequently everyone in the groups should master. Students can complete more than one role at a time, if necessary. Possible roles include the following:

Coach, student guides the group in the problem solving by facilitating the communication, motivate group members and if conflict case, he or she decide what strategy to choose or algorithm to apply, shifting the dialogue focus toward program structure. The coach asks the other groups' coaches or/and the teacher for assistance, if necessary.

Theorist, student, who can easily explain of the concept discussed, searches quickly for resources, and reminds the group of the steps to follow in problem solving.

Programmer, student sits, in front of the computer, typing the code program properly using Intellij and all group members read and correct or share an idea.

Moderator, student watches the group and gives feedback. He or she looks for behaviors to praise. The student encourages all group members to participate in the discussion and assist one another. He or she evaluates how well the group has worked together and gives suggestions for improvement.

Secretary, during the whole-class wrap-up, the student write on the pedagogical notebook PN all the details from the problem statement to the solution: noting the main ideas the group shared, questions the group has generated, teachers' answers…

Planner, Student sets the timer for each part in the lab session and lets the group know when it is time to move on to the next task. He or she makes sure everyone participates and only one person talks at a time (the teacher might do this instead of students).

The teacher's roles

Besides designing the course, tutorials and Lab sessions and exercises sheets, the teacher also design the timing about exercises to be solve in each lab session according the courses' progress, and also reminds the groups, during the 90-minute lab sessions to swap roles every finished exercise. Teacher can help by actively listening to students' conversations and if necessary clarifying situations, modeling strategy usage, explaining errors after or before debugging, sometimes correcting the code or giving tips about possible solutions, encouraging students to participate, and modeling a helpful attitude. At the end of each exercise, the teacher selects the best solution or final code in the perfect situation or suggests the worst code to be corrected, or to found the mistakes and at last taking notes on the PN. It is expected that students will need assistance learning to work in collaborative groups, implementing the strategies, and mastering academic content.

3.3 The semester' content and resolution strategy

The content

The Java lab sessions is conducted as shown in Table 2. The exercises' sheets are given at the start of each chapter that permits the students to prepare at least one or two exercises solutions before each groups' meeting in the lab session classrooms or other places. The teacher at the beginning of the class checks every student's solution and gives scores. Because lack of time, we select the most important exercises to solve in class and let the students do the rest outside of classroom time.

To foster the collaborative learning, the teacher might give highlights or mention the main idea of the checking skills and knowledge of Java programming. In the same date, we make an announcement of project groups' assignment and collaborative learning conditions to do during 7 weeks.

A project consultation involving collaboration among the students to make a code of simple software with user interface is conducted at the 14th week. The students learn advanced OOP in the latter half of the semester.

week	Session content
1 st	Groups determination, Intellij installation and first uses
2 nd	Sitting up groups & Uses java terminology 1
3 rd	Uses java terminology 2
4 th	Java Class 1
5 th	Java Class 2
6 th	Heritage & Polymorphism 1
7 th	Individual CC1 & Project groups Assignment
8 th	Heritage & Polymorphism 2
9 th	Abstract class & Interfaces 1
10 th	Abstract class & Interfaces 2
11 th	Graphic Interface 1
12 th	Graphic Interface 2
13 th	Applet
14 th	Individual CC2 & Project Consultation

Table 2: The Java lab sessions content per week.

The resolution strategy

Figure 1shows an overall scenario for learning activities in case the given exercise is being to be solved in class (solve at minimum two exercises per session: 20 min per exercise), otherwise we jump to phase 3 (the student prepare exercises before class to gain time and we correct more than two exercises in class). That involves:

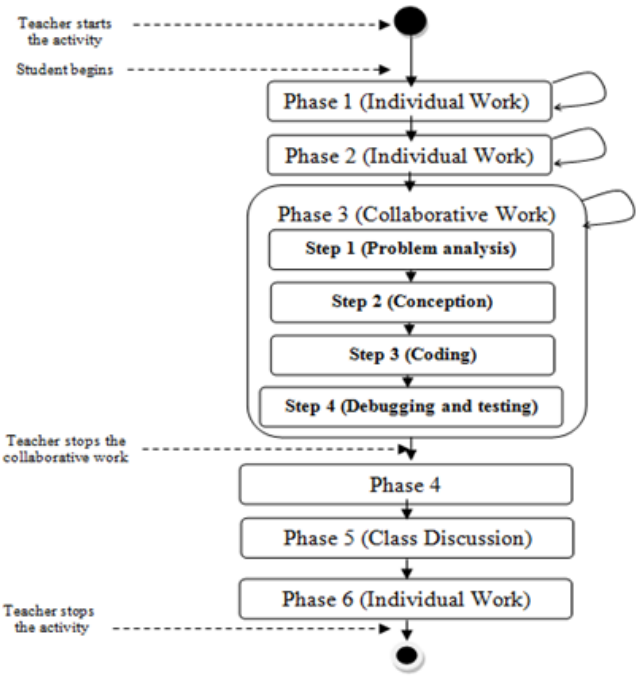

Figure 1: Global scenario for learning activity Problem-solving phases for programming.

Phase 1 (Individual work): students try to read and understand the exercise statement after that the teacher can explain the exercise's purpose, give some details and set up the timer for 20 min.

Phase 2 (Individual work): students adopt a strategy of resolution and try to solve the problem partially or totally.

Phase 3 (Collaborative work): it involves 4 steps:

3.1 Discussion (Problem analysis)

3.2 Choice of resolution strategy & writing algorithms (Conception)

3.3 Implementation (Coding)

3.4 Debugging and testing the final program.

Phase 4: the teacher stop the collaborative work, selects a code from any group and ask the group's coach to write it or just fragments on the whiteboard.

Phase 5: the teacher starts a discussion with the students about the code, finding bugs, and modifies it together, then ask the students to type it again on machine to display the results.

Phase 6 (Individual work): writing notes.

4 Results and discussion

So far, we have recorded 31 collaborative sub-groups for the two semesters. The following step is, for each one of them, to trace, in categories their characteristics, communication, benefits, disadvantages and other points of interest such as the technologies used. For instance, Group organizing refers to role division of various types that offer features of organizing a group. Collaboration environments refer to places or environments that allow students to study together in the same physical area that provides the appropriate infrastructure to support the collaboration environment for exchange and interaction used with a common purpose with Github in distance learning. Communication refers to different ways of supporting discussing and interaction between members in a group. Web conference and Webinars are synchronous ways of communication using internet and social network Facebook. Of course, there are other ways (for example asynchronous) that can thoroughly support groups' communication; like telephone messages.

4.1 Survey and data analysis

The online survey questionnaire was sent by this link https://app.evalandgo.com/s/?id=JTk1byU5NG4lOTklQUI=&a=JTk1ayU5OG0lOUIlQTk= via the students' Facebook group to about two promotions of 108 2nd year informatics students. The survey was conducted for the first promotion at the second half of January 2017 and at the beginning of February 2018. The survey was done in French language. The students had only one month to respond to the questionnaire, they were not graded. It was optional to complete. The responses were not anonymous at the time that they were submitted but were identified during data analysis. The ongoing collection of student responses was purposefully tied to the content questions so as to get the whole picture of the student experience in the Lab session Collaborative learning.

During this period, in total 97 questionnaires were received with response rate of about 71.59%. After initial analysis of the total responses, 31 of them were judged to be incomplete and were dropped from the analysis, bringing the number of surveys to be analyzed to 66 responses.

Survey consisted of multiple-choice, 4-point Likert scale, and short-answer questions that asked students about various aspects of collaborative learning lab session new design. Briefly, it contained 109 questions divided on 8 parts:

Part 1: Personal Information
Part 2: Use of E-Learning Platforms (Distance Learning)
Part 3: OOP Course
Part 4: Students' perceptions of collaborative work within the group
Part 5: Students' perceptions of group design for Lab session
Part 6: Student Perceptions of Role Division in a Group.
Part 7: Perceptions of students in Collaborative Production
Part 8: Students' Perceptions of the quality and progress of their learning during Learning Scenario Activities (Global Pedagogical Scenario).

Part 1: Personal Information. For the part 1, 44.44% of the responders were females and 55.56% were males. All students had BAC diploma and were between 20 and 25 years old except for one student who was over 25 years old. 53.33% attended the lectures, tutorials and lab sessions where the rest attended tutorials and lab sessions only. We noted that 82.22% of the students prefer to do their important homework outside of the university and the most of 17.78% stay in the university's library. 47.73% study for less than one hour per day, 38.64 % between 2-3 hours and 13.64 % for more than 3 hours. We asked the students about how they contact their teachers as multiple-choice question, 88.64 % answered that they do it directly after the teacher finishes the lectures, or tutorials or lab sessions and/or by social media about 45.45%, 31.82 % by emails and only 6.82% via the class's representative. Another multiple-choice question about the use of the internet: 88.84 % use it in search of exercises and solutions of different course, 29.55 to do tests online, 50% were looking for courses documentations, 20.45% to see their emails, and 25% to do other things.

Part 2: Use of E-Learning Platforms (Distance Learning). Regarding the 2nd part that was 8 multiple-choice, 4-points Likert scale, and one short-answer format as mentioned in Table 3 (see Appendix (A)). We want to know from this section if the students visit e-learning platforms to study and what they do exactly. Apparently, only half of students know the e-learning platforms and 65.91of them believe in its effectiveness for their learning. Generally, they wished for changing content and format of these platforms. For ICT courses taken in first-year, 81.82% took 1-2 courses, 4.55% from 2-4 courses, 4.55% took more than 4 courses and 9.09% haven't took any course.

Part 3: The OOP course. The OOP course design was a 6 of 4-point Likert scale, 1 multiple-choice and 3 short-answer questions format. The students, 65.79%, know that there is no OOP course on the e-learning platform, where the others don't. The division of the pedagogical content on courses, tutorials and lab session was satisfiable for 70.27% of the students, 27.03% disagreed and only 2.7% strongly disagreed. Then 48.65% of the students were strongly agreed to receive the course's content in video format, 21.62% agree, and the rest disagreed. All the students preferred to receive the contents of all sessions (courses, tutorials and lab session) before class, argues that this act will help them to prepare well and have more ideas about the classes 'content, and wish to have it online with tests and exercises' solutions with details, and know what their mates were reading, which courses. While the majority of students accessed to this part to complete all the questions (90–100% of students), a small percentage (5–10%) of students opted to not complete the 2nd part, particularly the short-answer questions only in two weeks. For better response rate, we should divide the survey on short-surveys and ask students to complete in different time of the whole experience. As few students generally wrote a few words for the short-answer questions, it is unlikely that any given short survey takes less than 5 min to complete.

Part 4: Students' perceptions of collaborative work within the group. It consisted of 7 questions of 4-points Likert scale; that allowed students to evaluate their opinions and understanding of the collaboration, coordination and communication in the group with all members.

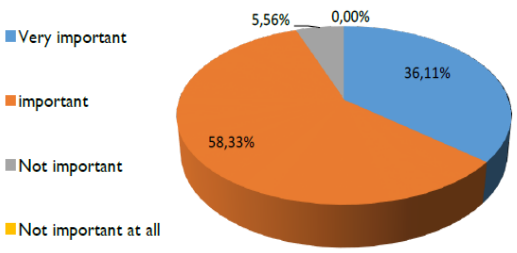

Figure 2: Students' Performances in Collaboration within the group.

Three principles questions were asked within "I would qualify my group during this practical work from a point of view". It was regarding the student experience in the collaborative work were based on the students' interactions with each other's inside the group and emerging themes observed during lab sessions. The data set shown in Table 4 (see Appendix (A)) that all students were happy and agreed of the idea of collaborative group work excepting for 5.56% were disagreed and preferred working solo (see Figure 2).

In coordination theme, tasks' assignment between groups' members were very important at 27.78%, important at 58.33% and not important at 13.89%. Just for information, tasks' division is different from roles' division; roles could be divided into tasks done by one or more than student. This might be more clarifying in part 6. As shown in Figure 3, the crucial element in the whole process was the time management including for the teacher, students in some steps in problem-solving phases overflow time allocated to finish the exercise when they (63.89%) could not decide about a definite point especially discussions and exchanging tasks (55.56% said that is not effective).

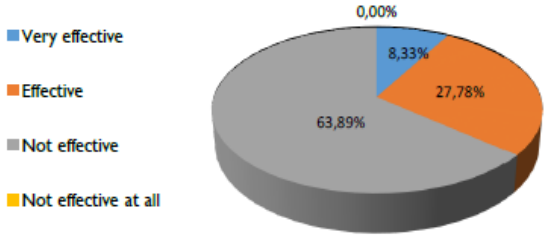

Figure 3: Students' Performances in Coordination (Time management).

They couldn't move on to the next step till it is not resolved. So, the Planner should be more lucid and specific in notifying the group members. This is why the teacher always notices all the groups when they are running out of time. To gain time, we allow groups to separate into subgroups (25.00%, 41.67% confirm its importance), more than one student take over one or more tasks. Some students (33.33%) chose to finish alone their tasks assigned. Communication was the vessel conductor in the group, students enjoy listening among the groups' members to learn and understand the context (Very effective at 33.33%, effective at 58 .33% and not effective at 8.33%), as seen in Figure 4.

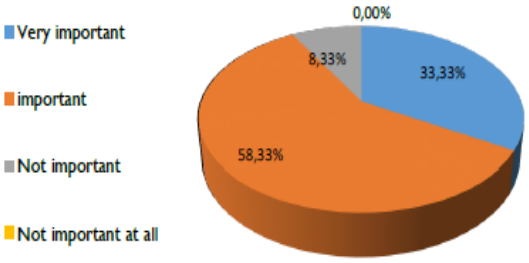

Figure 4: Students' Performances Listening among the groups' members.

When students agreed to divide simple tasks, help and depend on each other, they already know exactly what to do to finish it, as a result fast execution of tasks was very successful (Very effective at 36.11%, effective at 61.11% and not effective at 2.78%).

Part 5: Students' Perceptions of Group Design for Lab Session. This part of the survey is composed of 6 questions of 4-point Likert scale and one short question format (rated from Totally agree, Agree, Disagree, Strongly disagree); it is about groups' design and conditions for group members' adhesion. In Table V (see Appendix (A)), students consider 3 to 5 members in the group was appropriate (47.22% totally agree, 25.00% agree) due to the complexity of problem-solving and short time to realize it. We adopted the flipped-class

approach in preparing and solving exercises outdoor class to spend more time in rich discussions at lab sessions. They students know that there is not one way to solve a programming exercise but there is always an optimal one in term of running time, memory allocation, the right algorithm and minimum code lines. 80.55% of students get connected with their partners in the same group, 19.44% were not may be because in few groups some males and females were very shy (7 balanced groups, 2 groups of females and 2 groups of males). Conflicts in those groups were usually more balanced. Thus, the data set confirms that they prefer choosing their group mates and since we don't know the background of the students, it was not possible to let students select their group mates. The greater part of students wished for different academic level groups in order to lower academic level students get help from higher academic level students. 8.33% only which represents almost higher academic level students wants more competitive collaborative groups at the same time.

Part 6: Students' Perceptions about the roles' distribution within a group. The fundamental preoccupation in computer science pedagogy is the improvement of computational thinking skills where all students are involves in this process from every step in solving-problems to implement applications. Students were asked about the impact of the roles proposed for the entire process to perform in collaborative groups for programming assignments; its benefits in facilitating the project's realization. Also, they were asked about further elements that reinforce the group work and what does it take to achieve each role? Table VI (see Appendix (A)) shows that they were agreed (27.78% totally agree, 72.22% agree) that it is facilitating their achievement by helping each other to understand the problems and solutions. Besides developing computational thinking skills (100%), it attributes them to build trust (41.67%), to arise their commitment (44.44%) and motivation (47.22%). It was evident for students to change and switch roles in terms of tasks (25.00% totally agree, 55.56% agree), skills required and also to gain abilities in mastering all roles (16.67% disagree, 2.78% strongly disagree). Students must be engaged in front of all members and fully responsible on his simple goal to realize the group's goal, Fig. 5 (see Appendix (A)) shows that 16,67 % protest doing other roles to accomplish the programming tasks, we should investigate more in order to determinate witch roles could be defined to those defined before. Two roles (programmer, secretary) were done only by 13.89 % and 27.78 % of students respectively. Each one of these roles has its special criteria, required coding skills and abilities like formal pertinent writing and students were less confident and afraid of making errors and bugs. It can be explained that they have poor abilities to do such roles. 47.22 % of students were holding the role of coach imply that they were interacting more in the collaborative learning. Whereas the role of moderator was hold by 44.44 % that characterizes the engagement to encourage participation of all group members, and theorist were 44.44 % that provide confirmation that they understand well the concept being discussed and could easily explain it for others and only 41,67 % for planner role which reflects organizational feature in problem-solving for group learning. All Students might have already some individual skills or knowledge and could foster getting others by practicing, discussing and watching others doing it because they like to ask spontaneously their peers more than the teacher. Consequently, all percentages join the results shown above. As last, two short-questions format for this section, we asked student about benefits and inconveniences. Many students wrote that they were learning fast, getting great ideas form discussion, correction knowledge about wrong concepts' comprehension, working effortlessly without pressing, saving more time for writing algorithms and codes. Little minority seem to dislike this type of group work in distance because of difficulties in unreliability of some groups members, scheduling meetings, and while students divide up the programming task for a project, each do a code fragment, and then combine all pieces into a one finished program.

Part 7: Students' Perceptions in Collaborative Production. That concern what was used in collaborative production as tools and technologies. Table 8 indicates some students' answers performance. It involves different components:

On the proposed programming environment: in this section, we demand students about the use of that was chosen in first place for several reasons principally because it is intuitive and intelligent as environment and can help beginners to master programming and lets them to quickly and easily write and change the code. Students shows their resistance and refuse strongly (only 13.89% said that is effective, 44.44% not effective and, 41.67% said it is not effective at all) this new environment because they preferred working with familiar platforms like Eclipse or Netbeans and that Intellij IDEA is not easy to use (16.67% claim that is easy to use and the rest 83.33% assumed the opposite). For the next question they argue about the choice of the programming environment depends on (the tool's mastery 16.67%, the tool's ease of use 38.89%, the community of developers 30.56%, The needs of the work to be solved 13.89%) that explain why they would not exploit Intellij IDEA). Since the student viewed the proposed programming environment as an instrument for programming rather than only the required lab session which they had to pass the computer examination. Besides, they administered various strategies as well as a long period in practicing outside their class for developing the better learning techniques.

On tools used for communication and publishing tools: The data collected and evaluated about student use of communication and publishing tools confirmed that the tools were very helpful and appreciated by students for collaborative group in distance learning. In class, the tools were not employed for two reasons: 1) no internet available in the lab session, 2) they do not need it because only one computer was assigned to each group. At first, we must notify that categories, who possess personal laptops, smart phone or other devices with

internet, declared that the tools' use was very important. (27.78% were totally agree and 58.33% were agree), they needed to create Facebook groups to communicate via videos or calls and exchanging messages, discussing or at least to schedule meetings sessions and face-to-face appointments (63.89%), Github were used for displaying code fragments (22.23%). The other category were only 13,88% were connected to their colleagues by phone calls and proclaimed that Github was very sophisticated tool, consummate time to master and it deviate them from the principal purpose of programming.

On the necessary use of Pedagogical Notebook: The students may take notes, as a complementary learning activity for programming assignment. We decide to make it a good habit by requesting every group. Students responds (13.89% were totally agree, 63.89% were agree, 16.67% were disagree and 5.56% strongly disagree). That can be explicable: taking notes is a complex process from comprehension, selecting pertinent information in limited time. Other explanation, students replaced in most of the time PN as paper support by video, audio and images supports using its smart phones, justifying this action that as electronic support has better writing, and don't contain errors especially programming codes by answering the question:" Is a electronic format of PN better to handle than the paper format?", (36.11% were totally agree, 27.78% were agree, 30.56% were disagree and 5.56% strongly disagree).

About the lab session room: To enable students having modern teaching perspectives in lab rooms that allow collaborative learning, they were asked if they prefer working on their laptops instead of machines in the room. The majority of the students preferred their laptops (66.67% were totally agree, 16.67% were agree) to avoid problems like virus transferred with movable devices, and 16.67% were disagree because they do not have personal computers. Other questions regarding Classroom Structure or Computers Disposition form and the ideal arrangement for group work. They were all disagreed about the fixed computers tables and preferred (circle form 33.33%, the V form 52.78%, square form 11.11% and 2.78% wished for other flexible forms). Wi-Fi high-speed Internet access, interactive or intelligent board were the most wanted technologies for lab session. At minimum, students needed suitable spaces with audiovisual components for short demonstrations of tools installation, codes debugging, and lectures delivered during the session. They require also movable tables, computers and networks for higher collaborative organization supplies performance and an impressive room design to satisfying learning goals.

About the teacher's accompaniment and assistance for lab session: This might be divided on two: face-to-face and distance support. Students were thoroughly convinced that is the most important and necessary part in learning process. They were satisfied at 94.45% and 5.56% required for more than one teacher in lab sessions, they explain that the teacher should know the progress of each group globally and each student in detail in a special sheet assessment available for consulting every session.

Part 8: Students' Perceptions of the quality and progress of their learning during Learning Scenario Activities (Global Pedagogical Scenario. The students were generally satisfied but it reveals some weakness in different phases. The final set of data related to the global scenario proposed above rejoin that is very large category of students faces enormous difficulties in different steps in problem-solving in engineering sciences, especially in first steps e.g. analysis and designing (writing algorithms). Consequently, it is reasonable to think that students want more interaction with their peers in such phase where they are meeting problems. Moreover, some phases are believed to excel at some tasks more so than others do, e.g., phase 1, phase 2. A minor misunderstanding leads to a greater number of defects. The students were interested in-group discussion by listening attentively.

	Totally Agree	Agree	Disagree	Strongly disagree
Phase 1	27.78%	41.67%	27.78%	2.78%
Phase 2	13.89%	50%	33.33%	2.78%
Phase 3.1	38.89%	52.78%	8.33%	0 %
Phase 3.2	41.67%	38.89%	19.44%	0%
Phase 3.3	30.56%	50%	16.67%	2.78%
Phase 3.4	47.22%	50%	0 %	2.78%
Phase 4	17.14%	31.43%	48.57 %	2.86%

Table 5: Some students' responds performance for part 8 questions.

They collaborated in working with their peers for getting main idea of the exercise. We compared the two first lines with the two next ones, it clearly obvious that the positive and significant role of collaborative work to overcome with discussion the lake of understanding where a few students were disagree especially the best; they preferred not wasting time to explain to their peers, time they need to move on and start coding. It might be explained that they do not esteem the notion of sharing and acquire poor collaborative skills where the lower academic level students appear passive peers by holding the others back. Given the magnitude of the percentage in the last two steps in phase 3 regarding coding and debugging, however, we can reasonably assume that the lack of significance is not related to insufficient data, rather, it is because the behavior of most students in the class is very close to watcher behavior across all sessions. It could be concluded that the collaborative work could alleviate their anxiety in coding step, which would cause their low programming learning achievement. We confirm this by recoding weakness of programming as a gap between theory and practice.

In summary, much effort was expending in collaborative programming but, in other hand, the improved communication skills, enjoying lab sessions and reporting confidence in programming ability were the highest gains.

4.2 Assessments of programming improvement

In the context of this study, we use learning to refer to the improvement in programming skill, not knowledge of an abstract concept. Three different assessments in lab session for students are done in the 7th and 14th week: Individual CC1, Individual CC2 and Project Consultation, we calculated individual learning gain as follows:

In individual lab session LCC1 and LCC2: The individual LCC1 and LCC2 assessments were graded using a 10-point rubric. We examined ability for programming by checking source files that each student made in order to measure their programming skill as individual learning gain ILG using formula (1)

$$\mathrm{ILG} = (\mathrm{LCC1} + \mathrm{LCC2})/2 \quad (1)$$

The Project Consultation PC: This phase endured at the 14th week. The students turned in final projects, presented it as groups. In the lab session room, they found helpful to use PowerPoint slides. After each presentation, the teacher used an oral questioning for each group and it was graded using a 5-point rubric, focusing on collaborative education outcomes. We can determinate three categories of groups: good, medium and lower performing groups depending on project achievement, active contribution of group's members, beating challenges found, etc. The CC score for Lab session LCC is calculated as (2) follow:

$$\mathrm{LCC} = \mathrm{ILG} + \mathrm{PC} + \mathrm{P} \quad (2)$$

P refers to a score using a 5-point rubric relative to the students' attendance. From the results, initials analysis has shown three different categories: High, Average, and low scores. We compared Initial Test, Lab session LCC, Tutorials TCC and the final exam FE.

4.3 Assessments findings

For a good interpretation the results of this study, we remember that every assessment is done differently. The final exam, which constituted 50% of the course grade, consisted of 5 simple questions and 4 exercises including: identifying Java vocabulary, tracing source code outcomes, and implement a code. The TCC grades represent 25% of the course grade equally with LCC grades. The TCC was managed differently from LCC; it consisted of the average of two TCC1 and TCC2 assessment based on implementing a simple code each on paper in 20 minutes period. The final course grade was calculated on the weighted percentage of all assessment activities (FE, TCC and LCC). Therefore, the main reason for comparing those results is to recognize the improvement in computational thinking and programming skills and style in collaborative learning. Fig. 6 displays the key predictors of performance in relation to assessment results in three levels. Starting with IT, at the beginning of the semester, students have low performance level (62.03% of Low performance), only 7.40% have High performance and 30.55% as Average performance. As can be seen, LCC results, correlated with collaborative programming, were positively significant.

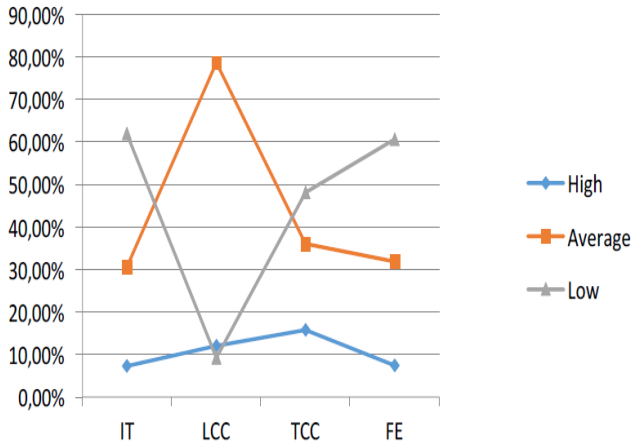

Figure 6: Improvement of students' results across different assessments.

The average level, which represented 78.70 % of the students, dominate the results with the lowest percentage 9.25 % of students in low level that shows a downward change. The percentage of the total number of students in Average performance has jumped highly from 30.55% to 78.70%. In High performance, it moved 7.40% to 12.05%. For TCC, we observe some drop in average and low levels with a small upward change in high level. Next in the transition from learning to the last assessment form, accompanied by teachers' surveillance and exam stress, we again observe poor and negative students performance across different levels.

5 Conclusion

This paper has investigated the educational effectiveness of a collaborative learning design in object-oriented programming language lab session at the university Mostafa Stambouli Mascara and students' satisfaction about different related issues. The results obtained allow us to claim that group design, global pedagogical scenario and especially roles division combined with technologies tools are the pedagogical methodology that produce in students. Furthermore, engagement, commitment and motivation of students increased at each phase of programming process.

Besides verifying that the global scores, the academic level increase was verified among the students that had an initial lower academic level. Collaborative learning groups exhibit new social skills in all students. Higher academic level students were the most resistant to collaboration at the beginning of the experiment. It was also observed in these groups that students with low academic level tended to be passive towards high academic level students. But the connection takes place only while the teacher intervenes to weld broken the groups and encouraged the participation of all group members. However, other students were more motivated and collaborative. The teacher assistance was very important, helping students groups to accept critical and constructive feedback reinforce their learning inside and

outside lab session room. The experiments also verified that in absence of the teacher, some groups were not effective outdoors lab session, and all members count on high academic level students to do all the work. By contrast, high academic level students impose their solutions for the other members without any discussion. In the same time, it conducts to unfinished discussion, where each member believed that he has the right point. Furthermore, dividing tasks and roles made the learning less difficult and more agreeable; students with high determination continue to work in groups in other courses since they build trust and solid membership. Other courses' teachers mention, in meetings, the good change of students' programming style who still works collaboratively. Selecting an appropriate delivery format becomes even more pronounced when the students are engaged in a collaborative learning environment. Designing a collaborative learning environment should include all element cited before. Obviously, learning becomes more driven by ICT tools. Facebook and Github are emerging technologies that supported students in the successful achievement of collaborative blended learning. Teachers inspected all source codes at the end of the experiment reported by data indicating that collaborative group has an affirmative influence on learning how to program, at least shows improving of students programming style. In sum, our results suggest that ICT tools must be added to the pedagogical methodology to increase the learning progress, but some new suggested tools faced by a large resistance and show opposite expectations. Until this stage, we have not discussed the choice of Java as a programming language because, firstly, it is a powerful language, then students had already studied the C language and it is mandatory in the minister official program. Since our students learned programming in first year and their lower performance level, we believe that we should prepare them as native programmers programming for second-year java collaborative programming for better results.

Limitations: The entire design is very exhausting practice from the beginning of the semester to the end, from initial tests to sending the final exam scores. So many difficulties demotivate both teachers and students in absence of minor ICT technologies, time allocated, networks, and management contesting often that conduct to discouragement for teachers and poor emerging skills for students.

Perspectives: Finally, working with limited available components infect on learning methodologies, thus it is an open issue for the future. Roles proposed for students in the pedagogical scenario should be studied carefully. The metacognitive abilities behind every task in every role are strongly related; it is essential to inspect and explain witch poor skills behind role were less picked. Our vision is to create a platform for learning Objet-Oriented programming language with all communication and visualization features that allows collaborative teaching and learning programming. We have to unite inspiring all students with clear and visible guidance to have equal opportunities to reach their full potential. Helping our students overcome their obstacles, in order to raise the level of participation in collaborative work, keep it moving forward in order to become more competitive in programming.

6 References

[1] J. Carter, P. Dewan, and M. Pichiliani, "Towards Incremental Separation of Surmountable and Insurmountable Programming Difficulties," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, New York, NY, USA, 2015, pp. 241–246. https://doi.org/10.1145%2F2676723.2677294

[2] D. Sleeman, "The Challenges of Teaching Computer Programming," *Commun. ACM*, vol. 29, no. 9, pp. 840–841, Sep. 1986. https://doi.org/10.1145%2F6592.214913

[3] P. H. Tan, C. Y. Ting, and S. W. Ling, "Learning Difficulties in Programming Courses: Undergraduates' Perspective and Perception," in *2009 International Conference on Computer Technology and Development*, 2009, vol. 1, pp. 42–46. https://doi.org/10.1109%2FICCTD.2009.188

[4] H. Belhandouz, "Teaching science in Algeria: pedagogical shortfalls and conflicts of meaning," *The Journal of North African Studies*, vol. 16, no. 1, pp. 99–116, Mar. 2011. https://doi.org/10.1080%2F13629387.2010.529655

[5] A. V. P. Arboleda, L. R. R. Mazuera, and G. S. Montemiranda, "Competences that facilitate the achievement of the objectives of an introductory programming course," in *2015 International Conference on Interactive Collaborative Learning (ICL)*, 2015, pp. 1007–1012. https://doi.org/10.1109%2FICL.2015.7318166

[6] A. Robins, J. Rountree, and N. Rountree, "Learning and Teaching Programming: A Review and Discussion," *Computer Science Education*, vol. 13, no. 2, pp. 137–172, Jun. 2003. https://doi.org/10.1076%2Fcsed.13.2.137.14200

[7] C. Baron and L. Baron, "Trois approches d'apprentissage collaboratif dans l'action pour soutenir le développement du leadership," *Humain et organisation*, vol. 1, no. 2, pp. 24–32, Nov. 2015.

[8] G. Alexandron, M. Armoni, M. Gordon, and D. Harel, "Scenario-based Programming: Reducing the Cognitive Load, Fostering Abstract Thinking," in *Companion Proceedings of the 36th International Conference on Software Engineering*, New York, NY, USA, 2014, pp. 311–320. https://doi.org/10.1145%2F2591062.2591167

[9] B. Isong, "A Methodology for Teaching Computer Programming: first year students' perspective," *International Journal of Modern Education and Computer Science(IJMECS)*, vol. 6, no. 9, p. 15, Sep. 2014.

[10] M. Hauswirth and A. Adamoli, "Teaching Java programming with the Informa clicker system," *Science of Computer Programming*, vol. 78, no. 5, pp. 499–520, May 2013.

https://doi.org/10.1016%2Fj.scico.2011.06.006

[11] G. Fesakis and K. Serafeim, "Influence of the Familiarization with 'Scratch' on Future Teachers' Opinions and Attitudes About Programming and ICT in Education," in *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA, 2009, pp. 258–262. https://doi.org/10.1145%2F1562877.1562957

[12] S. Mishra and S. Iyer, "Problem Posing Exercises (PPE): An Instructional Strategy for Learning of Complex Material in Introductory Programming Courses," in *2013 IEEE Fifth International Conference on Technology for Education (t4e 2013)*, 2013, pp. 151–158. https://doi.org/10.1109%2FT4E.2013.45

[13] A. Abdullah and T. Y. Yih, "Implementing Learning Contracts in a Computer Science Course as a Tool to Develop and Sustain Student Motivation to Learn," *Procedia - Social and Behavioral Sciences*, vol. 123, pp. 256–265, Mar. 2014. https://doi.org/10.1016%2Fj.sbspro.2014.01.1422

[14] J. M. Rodríguez Corral, A. Civit Balcells, A. Morgado Estévez, G. Jiménez Moreno, and M. J. Ferreiro Ramos, "A game-based approach to the teaching of object-oriented programming languages," *Computers & Education*, vol. 73, pp. 83–92, Apr. 2014. https://doi.org/10.1016%2Fj.compedu.2013.12.013

[15] H. Cigdem, "How Does Self-Regulation Affect Computer-Programming Achievement in a Blended Context?," *Contemporary Educational Technology*, vol. 6, no. 1, pp. 19–37, Mar. 2015.

[16] C.-H. Chen, K.-C. Wang, and Y.-H. Lin, "The Comparison of Solitary and Collaborative Modes of Game-based Learning on Students' Science Learning and Motivation.," *Educational Technology & Society*, vol. 18, no. 2, pp. 237–248, 2015.

[17] M. Maleko, D. Nandi, M. Hamilton, D. D'Souza, and J. Harland, "Facebook versus Blackboard for Supporting the Learning of Programming in a Fully Online Course: The Changing Face of Computing Education," in *2013 Learning and Teaching in Computing and Engineering*, 2013, pp. 83–89. https://doi.org/10.1109%2FLaTiCE.2013.31

[18] C. Z. Kertész, "Using GitHub in the classroom - a collaborative learning experience," in *2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2015, pp. 381–386. https://doi.org/10.1109%2FSIITME.2015.7342358

[19] T. Kasame, K. Pachoen, and A. Manit, "The use of engineering design concept for computer programming course: A model of blended learning environment," *Educational Research and Reviews*, vol. 11, no. 18, pp. 1757–1765, Sep. 2016. https://doi.org/10.5897%2FERR2016.2948

[20] G. Ion, A. Barrera-Corominas, and M. Tomàs-Folch, "Written peer-feedback to enhance students' current and future learning," *International Journal of Educational Technology in Higher Education*, vol. 13, no. 1, Dec. 2016. https://doi.org/10.1186%2Fs41239-016-0017-y

[21] A. Othman, A. Impes, C. Pislaru, and D. Wilson, "Virtual Lab for Teaching a Computer Programming Course," *Online Journal of Art and Design*, vol. 2, no. 4, pp. 42–54, 2014.

[22] G. Álvarez and L. Bassa, "ICTs and collaborative learning: a case study of a class blog for improving the writing skills of pre-university students," *RUSC. Universities and Knowledge Society Journal*, vol. 10, no. 2, p. 5, Jul. 2013. https://doi.org/10.7238%2Frusc.v10i2.1740

[23] J. Owolabi, O. A. Adedayo, A. O. Amao-Kehinde, and T. A. Olayanju, "Effects of Solo and Pair Programming Instructional Strategies on Students' Academic Achievement in Visual-Basic.Net Computer Programming Language," *GSTF Journal on Computing (JoC)*, vol. 3, no. 3, Aug. 2014.

[24] A.-B. Al-Azawei, "Evaluating the Effect of Arabic Engineering Students' Learning Styles in Blended Programming Courses," *Journal of Information Technology Education: Research*, vol. 15, pp. 109–130, 2016.

[25] B. Estácio *et al.*, "Evaluating Collaborative Practices in Acquiring Programming Skills: Findings of a Controlled Experiment," in *2015 29th Brazilian Symposium on Software Engineering*, 2015, pp. 150–159. https://doi.org/10.1109%2FSBES.2015.24

[26] S. Hamid, J. Waycott, S. Kurnia, and S. Chang, "Understanding students' perceptions of the benefits of online social networking use for teaching and learning," *The Internet and Higher Education*, vol. 26, pp. 1–9, 2015. https://doi.org/10.1016%2Fj.iheduc.2015.02.004

[27] S. M. Alyami and A. M. Alagab, "The Difference in Learning Strategies in Virtual Learning Environment and Their Effect on Academic Achievement and Learning Satisfaction for Distance Teaching & Training Program Students," in *2013 Fourth International Conference on e-Learning "Best Practices in Management, Design and Development of e-Courses: Standards of Excellence and Creativity,"* 2013, pp. 102–112. https://doi.org/10.1109%2FECONF.2013.40

[28] A. Azemi, M. Bodek, and G. Chinn, "Teaching an introductory programming course using hybrid e-learning approach," in *2013 IEEE Frontiers in Education Conference (FIE)*, 2013, pp. 1911–1913. https://doi.org/10.1109%2FFIE.2013.6685168

[29] M. E. Muuro, R. O. Oboko, and P. W. Wagacha, "Evaluation of Intelligent Grouping Based on Learners' Collaboration Competence Level in Online Collaborative Learning Environment," *The International Review of Research in Open and Distributed Learning*, vol. 17, no. 2, Mar. 2016. https://doi.org/10.19173%2Firrodl.v17i2.2066

[30] J. Dwarika and M. R. R. de Villiers, "Use of the Alice Visual Environment in Teaching and

Learning Object-oriented Programming," in *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists*, New York, NY, USA, 2015, pp. 14:1–14:10. https://doi.org/10.1145%2F2815782.2815815

[31] Y. Hayashi, K. I. Fukamachi, and H. Komatsugawa, "Collaborative Learning in Computer Programming Courses That Adopted the Flipped Classroom," in *2015 International Conference on Learning and Teaching in Computing and Engineering*, 2015, pp. 209–212. https://doi.org/10.1109%2FLaTiCE.2015.43

[32] N. Funabiki, Y. Korenaga, Y. Matsushima, T. Nakanishi, and K. Watanabe, "An Online Fill-in-the-Blank Problem Function for Learning Reserved Words in Java Programming Education," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, 2012, pp. 375–380. https://doi.org/10.1109%2FWAINA.2012.64

[33] H. Jonsson, "Using flipped classroom, peer discussion, and just-in-time teaching to increase learning in a programming course," in *2015 IEEE Frontiers in Education Conference (FIE)*, 2015, pp. 1–9. https://doi.org/10.1109%2FFIE.2015.7344221

[34] M. Husain, N. Tarannum, and N. Patil, "Teaching programming course elective: A new teaching and learning experience," in *2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE)*, 2013, pp. 275–279. https://doi.org/10.1109%2FMITE.2013.6756349

[35] F. Silvestre, P. Vidal, and J. Broisin, "Tsaap-Notes – An Open Micro-blogging Tool for Collaborative Notetaking during Face-to-Face Lectures," in *2014 IEEE 14th International Conference on Advanced Learning Technologies*, 2014, pp. 39–43. https://doi.org/10.1109%2FICALT.2014.22

[36] S. L. Wismath and D. Orr, "Collaborative Learning in Problem Solving: A Case Study in Metacognitive Learning," *Canadian Journal for the Scholarship of Teaching and Learning*, vol. 6, no. 3, 2015. https://doi.org/10.5206%2Fcjsotl-rcacea.2015.3.10

[37] J. Olivier, "Blended learning in a first-year language class: Evaluating the acceptance of an interactive learning environment," *ResearchGate*, vol. 37, no. 2, Oct. 2016. https://doi.org/10.4102%2Flit.v37i2.1288

[38] M.-I. Dascalu, C.-N. Bodea, A. Moldoveanu, A. Mohora, M. Lytras, and P. O. de Pablos, "A recommender agent based on learning styles for better virtual collaborative learning experiences," *Computers in Human Behavior*, vol. 45, pp. 243–253, Apr. 2015. https://doi.org/10.1016%2Fj.chb.2014.12.027

[39] G. Temperman, B. De Lièvre, P. Bossaert, and J. De Stercke, "Effets de la distribution de rôles selon le style d'apprentissage dans un environnement collaboratif à distance," in *TICE 2010*, Nancy, France, 2010, p. 6.

[40] J. Andriessen, M. Baker, and D. Suthers, Eds., Arguing to Learn: Confronting Cognitions in Computer-Supported Collaborative Learning Environments. Springer Netherlands, 2003.

[41] A. Zagalsky, J. Feliciano, M.-A. Storey, Y. Zhao, and W. Wang, "The Emergence of GitHub As a Collaborative Platform for Education," in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, New York, NY, USA, 2015, pp. 1906–1917. https://doi.org/10.1145%2F2675133.2675284

[42] F. J. Rodríguez, K. M. Price, and K. E. Boyer, "Exploring the Pair Programming Process: Characteristics of Effective Collaboration," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, New York, NY, USA, 2017, pp. 507–512. https://doi.org/10.1145%2F3017680.3017748

[43] P. Dillenbourg, Over-scripting CSCL: The risks of blending collaborative learning with instructional design. Heerlen, Open Universiteit Nederland, 2002.

[44] C. Armstrong, "Catalyzing Collaborative Learning: How Automated Task Distribution May Prompt Students to Collaborate," *E-Learning and Digital Media*, vol. 7, no. 4, pp. 407–415, Dec. 2010. https://doi.org/10.2304%2Felea.2010.7.4.407

[45] K. Prahl, "Best Practices for the Think-Pair-Share Active-Learning Technique," *The American Biology Teacher*, vol. 79, no. 1, pp. 3–8, Dec. 2016. https://doi.org/10.1525%2Fabt.2017.79.1.3

[46] I. N. Umar and T. H. Hui, "Learning Style, Metaphor and Pair Programming: Do they Influence Performance?," *Procedia - Social and Behavioral Sciences*, vol. 46, pp. 5603–5609, 2012. https://doi.org/10.1016%2Fj.sbspro.2012.06.482

[47] S. Gross and N. Pinkwart, "Towards an Integrative Learning Environment for Java Programming," in *2015 IEEE 15th International Conference on Advanced Learning Technologies*, 2015, pp. 24–28. https://doi.org/10.1109%2FICALT.2015.75

[48] B. J. Parra, "Learning strategies and styles as a basis for building personal learning environments," *International Journal of Educational Technology in Higher Education*, vol. 13, no. 1, Dec. 2016. https://doi.org/10.1186%2Fs41239-016-0008-z

[49] G.-H. Hwang, B. Chen, and C.-W. Huang, "Development and Effectiveness Analysis of a Personalized Ubiquitous Multi-Device Certification Tutoring System Based on Bloom's Taxonomy of Educational Objectives," *Journal of Educational Technology & Society*, vol. 19, no. 1, pp. 223–236, 2016.

[50] A. H. Seyal, Y. S. Mey, M. H. Matusin, N. H. Siau, and A. A. Rahman, "Understanding Students Learning Style and Their Performance in Computer Programming Course: Evidence from Bruneian Technical Institution of Higher Learning," *International Journal of Computer Theory and Engineering*, vol. 7, no. 3, p. 241, 2015.

[51] C. Ferraris, A. Lejeune, L. Vignollet, and J.-P. David, "Modélisation de scénarios pédagogiques collaboratifs." 28-Jun-2005.

[52] G. Paquette, F. Crevier, C. Aubin, J. Rocheleau, C. Paquin, and M. Léonard, "Méthode d'ingénierie d'un système d'apprentissage," *Revue Informations In Cognito*, vol. 8, p. 1997, 1997.

[53] C. Yáñez-Aldecoa, A. Okada, and R. Palau, "New learning scenarios for the 21st century related to Education, Culture and Technology," *RUSC. Universities and Knowledge Society Journal*, vol. 12, no. 2, p. 87, Apr. 2015. https://doi.org/10.7238%2Frusc.v12i2.2454

[54] J.-P. Pernin and A. Lejeune, "Dispositifs d'apprentissage instrumentés par les technologies : vers une ingénierie centrée sur les scénarios," presented at the Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et de l'Industrie, 2004, pp. 407–414.

[55] W. Debabi and T. Bensebaa, "Using serious game to enhance learning and teaching algorithmic," *Journal of e-Learning and Knowledge Society*, vol. 12, no. 2, 2016.

[56] N. Mamat and N. Yusof, "Learning Style in a Personalized Collaborative Learning Framework," *ResearchGate*, vol. 103, pp. 586–594, Nov. 2013. https://doi.org/10.1016%2Fj.sbspro.2013.10.376

[57] A. R. Pacios Lozano and G. Bueno de la Fuente, "Teamwork and leadership in a virtual learning environment," *RUSC. Universities and Knowledge Society Journal*, vol. 10, no. 2, p. 112, Jul. 2013. https://doi.org/10.7238%2Frusc.v10i2.1452

7 Appendix (A)

	Absolutely Yes	May be	I do not	Not at all
Do you know the E-learning platforms	27.27%	22.73%	20.45%	29.55%
Do you think that the online courses help you in your learning	31.82%	34.09%	15.91%	18.18%
	Course format	Reading list	Online discussion	Online tests
Do you think that the content of these courses must be changed and / or must contain other things	75%	18.18%	59.09%	43.18%
	No course	1-2	2-4	+ 4
How many courses have you studied in ICT	9.09%	81.82%	4.55%	4.55%

Table 3: Students' responds performance for part 2 questions.

	Very important	Important	Not important	Not important at all
Collaboration within the group	36.11%	58 .33%	5.56%	0.00%
	Very effective	Effective	Not effective	Not effective at all
Coordination within the group: 1. Tasks' assignment	27.78%	58 .33%	13.89%	0.00%
2. Time management	8.33%	27.78%	63.89%	0.00%
3. Partition into subgroups	25.00%	41.67%	33.33%	0.00%
	Very important	Important	Not important	Not important at all
Communication within the group: 1. Listening among the groups' members	33.33%	58 .33%	8.33%	0.00%
2. Fast execution of tasks	36.11%	61.11%	2.78%	0.00%
3. Exchanges of ideas, tasks	16.67%	27.78%	55.56%	0.00%

Table 4: Performance for part 4 questions students' responds.

	Totally agree	Agree	Disagree	Strongly disagree
I consider the students' number per group suitable.	47.22%	25.00%	27.78%	0.00%
I get along with my partners in the same group.	33.33%	47.22%	19.44%	0.00%
When the group meets a conflict case, it is generally easily to overcome.	16.67%	52.78%	30.56%	0.00%
I prefer choosing my group's mate	80.56%	19.44%	0.00%	0.00%
I prefer students' same academic level in each group	8.33%	27.78%	61.11%	2.78%
I prefer students' different academic level in each group	16.67%	58.33%	22.22%	2.78%

Table 5: Performance for part 5 questions students' responds.

	Totally agree	Agree	Disagree	Strongly disagree
Does the distribution of roles facilitate the realization of collaborative work?	27.78%	72.22%	0.00%	0.00%
What criteria could strengthen the group work?	44.44%	41.67%	47.22%	100%
Do roles change according to the nature of the work requested?	25.00%	55.56%	16.67%	2.78%
I got involved significantly in my role	41.67%	47.22%	16.67%	0.00%
How do you evaluate the involvement of your classmates in their roles?	19.44%	58.33%	25.00%	0.00%

Table 6: Some students' responds performance for part 6 questions.

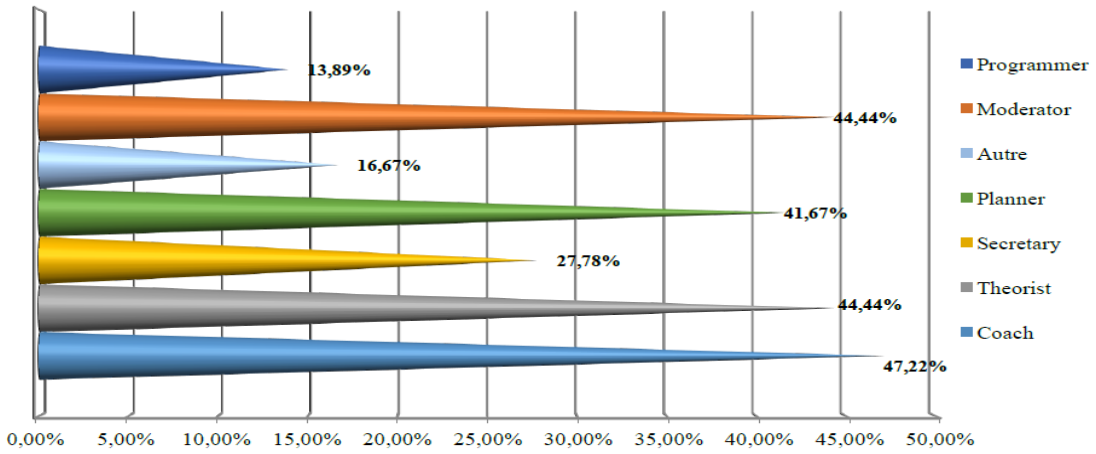

Figure 5: Roles' percentage held by students.