Received: 10 March 2021 | Revised: 29 May 2021 | Accepted: 10 July 2021

DOI: 10.1002/cae.22447

RESEARCH ARTICLE

WILEY

Using flipped classroom and peer instruction methodologies to improve introductory computer programming courses

Juan Ruiz de Miras[1,2] | **José R. Balsas-Almagro**[2] | **Ángel L. García-Fernández**[2]

[1]Software Engineering Department, University of Granada, Granada, Spain

[2]Computer Science Department, University of Jaén, Jaén, Spain

Correspondence

Juan Ruiz de Miras, Departamento de Lenguajes y Sistemas Informáticos, ETS Ingenierías Informática y de Telecomunicación, Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain.

Email: demiras@ugr.es

Funding information

Universidad de Jaén, Grant/Award Number: PID22_201617

Abstract

Learning to write computer programs and engaging students in introductory programming subjects are difficult tasks. This study presents the results obtained when using flipped classroom and peer instruction methodologies in an introductory course of programming. Before the lecture, out-of-class study and on-line homework questionnaires were applied, and peer instruction was used in the classroom to encourage students' participation and discussion, assisted by an on-the-fly assessment app. The main academic results, the degree of involvement in the course activities, and the students' feedback were collected and compared with similar data from previous academic years. Significantly better pass ($\chi^2 = 12.94$, $p < .001$) and dropout rates ($\chi^2 = 7.08$, $p < .01$) were obtained when using flipped classroom and peer instruction. Laboratory attendance also improved significantly ($\chi^2 = 27.62$, $p < .001$). Our results showed significant positive correlations between the work performed at home and the grade obtained in the final exam ($\rho = 0.51$, $p < .001$). According to the students' opinions, class preparation at home increased significantly ($\chi^2 = 46.59$, $p < .001$) by 33% due to the new methodologies used. These results, based on a sample of six offerings of the same programming course, suggest that the application of flipped classroom and peer instruction improves the quality of the autonomous work of the students as well as their involvement in the lectures, and therefore, their knowledge acquisition improves.

KEYWORDS

computer-aided instruction, flipped classroom, mobile learning, peer instruction, software engineering education

1 | INTRODUCTION

In the last 20 years, professors have increased the use of technology as a way to extend and enhance students' understanding. One of these strategies is the flipped classroom methodology, which consists of providing the students with instructional resources in advance, so that they can access and study them outside the classroom before the actual session. Then, the lecture time is devoted to working on the contents in greater depth with the support of the professor, clarifying those points that are still unclear for the students and identifying common mistakes. The interaction between the students in the classroom is a key point of this methodology to engage them in active learning [28]. In this way, students are prompted to explore knowledge through inquiry and collaboration, promoting a lively interaction within the classroom, where the professor acts as a mentor instead of a controller. As a result, positive effects are achieved on the students' knowledge, skills, and engagement [23].

Peer instruction is a teaching methodology that enables a student-centered learning environment by replacing some of the class time that used to be devoted to professor explanations with a student-focused activity. This activity consists of using a set of carefully designed questions to engage students in learning. Students discuss and analyze these questions in small groups and answer them using any kind of electronic device that enables both full participation and fast review of the answers [6,22].

Our programming course is delivered in the second semester (15 weeks) of the first year in the undergraduate Software Engineering diploma at our university (the students are majors). Our university is ranked at the positions 700–800 (world) and 23–30 (country) according to the 2020 Academic Ranking of World Universities (http://www.shanghairanking.com). Specifically for the Computer Science field, the ranking positions for our university are 201–300 (world) and 3–6 (country). Our university is public and the students mostly belong to middle-class families. Until the 2016/2017 academic year, this programming course was taught using professor-centered lectures for theory classes (2 h/week, 30 h in total) and programming assignments to be prepared at home before the laboratory sessions (2 h/week, 30 h in total). Additionally, at the end of each lesson, several theory hours were dedicated to either problem-solving sessions (students were invited to volunteer) or collaborative work discussing exercises in small groups. Together with the final written theory exam, students had to take two practice tests (one in the middle of the semester and the other one at the end) to assess the progression of their programming skills on the computer.

Although that variety of activities was planned to facilitate student learning, the course had been showing academic results below the expectations of its professors for many years, especially in the final exam grades and the dropout rate (i.e., the number of students who decide not to take the final exam). This is a common place for introductory programming courses because learning to program a computer is a hard task [14,25]. After attending a workshop on the flipped classroom and peer instruction methodologies [21], the three professors of the course decided to implement this methodology from the 2017/2018 academic year to help improve those low academic indicators.

The present study is based on a sample of 612 students over six offerings of the same introductory programming course, and shows the results of the application of flipped classroom and peer instruction methodologies, comparing them with those obtained in the previous offerings using traditional teaching.

2 | RELATED WORK

There exist several reviews on research into flipped classroom and its applications, both from a general perspective [1,7] and specifically devoted to Computer Science education [9]. The main conclusions of these reviews can be summarized in the following points: (1) flipping the class is a way to improve learning performance; (2) students show positive attitudes toward this approach; (3) the methodology enables student engagement; (4) there is a notable increase in the number of class discussions; (5) students improve their cooperative skills; and (6) they acquire better learning habits. Nevertheless, there are still some challenges that are encountered: (1) preparing the classes and developing the materials needed to flip the classroom require the professors to spend considerably extra amount of time; (2) sometimes, students are reluctant to accept a new methodology; and (3) there is a risk that class attendance could decrease, especially for courses with a high number of students.

Learning how to program is a difficult task, specifically because it is a process that has a tedious nature and lacks interpersonal interaction [2]. These factors do not help students to engage in this kind of subject [4]. Horton et al. [10] proved that flipped classroom helped improve the final grades in an introductory course on programming. Better motivation and a high implication of students in class activities as well as an increase in class attendance were achieved in an algorithmic thinking introductory course [29]. Flipped classroom was adapted to a collaborative environment based on Facebook groups for teaching programming [26]. The results of this study showed that the use of this approach has a positive effect on students' programming success and programming self-efficacy. Flipped classroom can also be combined with in-class mini-lectures of 10–15 min to better adapt the methodology to large classes and avoid the typical self-learning problems because no professor can provide complementary explanations [20].

In [12], Jonsson presents an approach based on screencast videos where in-class activities were centered on just-in-time teaching [24]. The results of this study showed that this approach allowed not only to increase by a third the number of students passing the course but also to double the share of students with good grades. Lepp and Tonisson [18] presented a combination of flipped classroom, collaborative learning, and peer review approaches for developing workshops in an introductory programming course. Most students acknowledged that the flipped classroom approach helped them to learn better and more than traditional lectures. Nevertheless, contradictory results were found in [5], where the flipped classroom offering of an introductory programming course achieved a similar dropout rate and average grade to the traditional course, but attendance at lectures decreased because students thought that the new in-class activities were not as helpful as the traditional lectures. A conclusion of this study was to incorporate peer instruction for delivering the classes in future offerings.

The primary motivation for flipping a classroom is to allow additional time for in-class activities, including active learning such as peer instruction. Several previous studies show that the main motivations to include this methodology in class are to improve students' learning and encourage them to become more involved in classes [19,8]. Porter et al. [27] performed a large study on the utilization of peer instruction in four computer science courses involving several professors. Their results showed that the adoption of this methodology reduces fail rates (the rate at which students fail to pass the course) by a per-course average of 61% compared to standard instruction. Another successful experience of using peer instruction in an introductory course to programming in MATLAB is described in [17], in which the pass rate increased and the dropout rate decreased when the new methodology was adopted. However, not all experiences are positive. For example, a study on applying peer instruction in a third-year computer science course aiming at increasing class attendance [16] showed that the students' feedback on the use of the new methodology was positive, although peer instruction itself did not prove to be a useful strategy for enhancing class attendance ratios.

Although significant research on applying flipped classroom and peer instruction to teaching programming has already been performed, almost all of the aforementioned studies showed results for experiments carried out with only one offering of the course and small student samples. Moreover, very little is known about the effect of using these two methodologies together in introductory programming courses. Therefore, the goal of the present study is to formally analyze this effect in a study covering a sample of six offerings of the course and more than 600 students.

3 | MATERIALS AND METHODS

3.1 | Subjects

We analyzed the academic results and the feedback of 612 students who were enrolled in the introductory programming course in six academic years, from 2013/14 to 2018/19. Students who enrolled in the course more than once were considered only once in our analyses; in those cases, we selected the data of the academic year in which the student took the final exam for the first time. For each academic year, only data from students who at least took one exam (either the final exam or one of the laboratory exams) were considered. We were interested only in those students who attended lectures regularly because they could really be affected by the methodological changes. In our experience, the students who regularly attended lectures were the ones who performed daily activities and finally took the exams. We selected those six academic years because all of them shared the same contents, the same set of assessment activities, and the same way of monitoring attendance. To avoid biases in the statistical analyses due to differences in the personal capacity of each student, and therefore in the average level of the students of each course, the individual grade obtained in the entry exam to the university (admission grade) was also compiled to incorporate it as a covariate in all statistical models. The course was taught by the same three professors during all the academic years considered in this study. Each one of the professors was in charge of one theory group as well as several laboratory groups. Table 1 shows the demographic data of our study participants.

3.2 | Methodology

From the academic years 2013/14 to 2016/17, we delivered the classes using traditional lectures in which the time dedicated by professors to explaining the concepts

TABLE 1 Study demographics

Year	Students	Admission grade[a] mean (standard deviation)
2013/14	99	7.34 (1.49)
2014/15	87	7.71 (1.89)
2015/16	100	7.58 (1.95)
2016/17	116	7.81 (1.57)
2017/18	119	8.28 (1.87)
2018/19	91	8.75 (1.82)

[a]The maximum grade that can be obtained is 14.

covered most of the class time. On the other hand, in the academic years 2017/18 and 2018/19, we implemented a flipped classroom approach together with peer instruction for in-class activities. As explained above, the contents of the course and their temporal distribution were similar for the six academic years analyzed, except for minor corrections due to bank holidays. Below, we detail the course organization and all the activities carried out for both methodologies.

3.2.1 | Traditional lectures

In the four academic years from 2013/14 to 2016/17, theory lectures (1 h per lecture, two lectures a week, 30 lectures in total) were delivered in a traditional way supported by material such as slides and source code examples. The students were provided with this material in advance. Laboratory sessions (2 h per session, one session a week, 15 sessions in total) were based on coding exercises that students had to complete at home before the session. In the laboratory, the professors discussed the solution of the exercises and attended to particular doubts individually. Additionally, students had to solve individually and on-site a simple but concise exercise about the session concepts in the last half hour of each lab session.

Five 1-h theory sessions (out of 30) in the course were devoted to discussing and solving coding and design problems in small groups. In addition, another six 1-h sessions (out of 30) were devoted to exercise solving, where students discussed their solutions in the classroom with professors and classmates.

Students had to take a final theory exam that was the same regardless of the professor and had the same structure each academic year (several short-answer questions and two programming exercises). In addition, students had to take two practice tests (one in the middle of the semester and the other one at the end). The questions in these lab tests were about the same concepts for each lab group and each academic year, although the test in each lab group had its own case study.

The final grade of the course (100%) was obtained according to the evaluation of the activities described above: final exam (40%), two lab exams (20% + 20%), lab exercises (10%), and individual class exercises (10%).

3.2.2 | Flipped classroom with peer instruction

Flipped classroom and peer instruction are the methodologies that we used in the offerings of the course in the academic years 2017/18 and 2018/19 (the academic year 2019/20 could not be considered in this study due to the different circumstances caused by the COVID-19 pandemic). These two methodologies were applied to the theory lectures. As in the case of the previous offerings of the course, the final theory exam was the same regardless of the professor and had the same previously described structure. Laboratory sessions maintained the same structure as in the previous course offerings. In the programming course, the theoretical concepts are directly exemplified in the C++ programming language. Code exercises to be solved in the laboratory are based on the concepts discussed previously in the theory classes.

Out-of-class activities

Professors provided the students with a learning-card 1 week in advance of each lecture so that they could adequately prepare and learn the corresponding concepts. Each learning-card contained all the information a student needed to correctly perform the out-of-class activities, namely:

- *Learning objectives of the session*.
- *Resources*: The material the students had to work with at home. The main learning resource we provided students with consisted of sections of a reference textbook that explained the concepts of the corresponding lecture. Additionally, students had access to the supporting slides that were used in previous years. These slides included code snippets and links to several full code samples that students could use to put the concepts into practice.
- *Tasks*: To guide and self-check the result of their autonomous study, a programming exercise was proposed to the students. This exercise was simple and concise, but focused on the new concepts of the lecture. Finally, students had to answer a short on-line multiple-choice test with three or four questions to self-check their level of comprehension. These out-of-class activities were organized through the on-line learning management system provided by our university.
- *Deadline*: Students had a deadline to answer the short questionnaire for each lesson. This deadline ended the day before the lecture.
- *Time for completion*: An estimation of the time that students were going to need to study the material and perform the out-of-class activities. The last question in each questionnaire asked students about the time they actually needed to study and complete the homework. In this way, the contents and complexity of each lesson were adjusted to the actual time that students needed to study them.

Before each class, professors spent some time checking the hit ratio (i.e., the percentage of right answers) of each question in the corresponding short multiple-choice test. In this way, professors were able to detect both the lesson concepts that had been clearly understood by students through their out-of-class study and those other concepts more difficult to comprehend. The in-class activities were then focused on these latter concepts [11].

In-class activities

Peer instruction methodology was used to guide the work in class, supported by the Socrative on-line service (https://socrative.com), which can be easily used from students' smartphones and laptops. We also tested some other software such as Kahoot (https://kahoot.com) for managing in-class questions interactively in several classes at the end of the academic year 2016/17. Finally, we selected Socrative because it allows professors to present questions to students more clearly, with a variable number of possible answers and with no time restriction. Due to the fact that the number of students per group exceeds the maximum number of concurrent users of the free version of Socrative, it was necessary to purchase the Pro version.

In each lecture, several questions about the most important and most complicated concepts (according to the answers obtained in the out-of-class tests) were asked to the students, and they could answer anonymously through Socrative using their own smartphones or laptops. Anonymity was very important for us because we wanted the students to answer what they knew exactly without any kind of pressure. Usually, two or three multiple-choice questions covered all the available time for 1-h classes. First, the professor presented the question to be answered individually. Students answered the question and the professor checked the hit ratio without revealing the right answer. Figure 1 shows an example

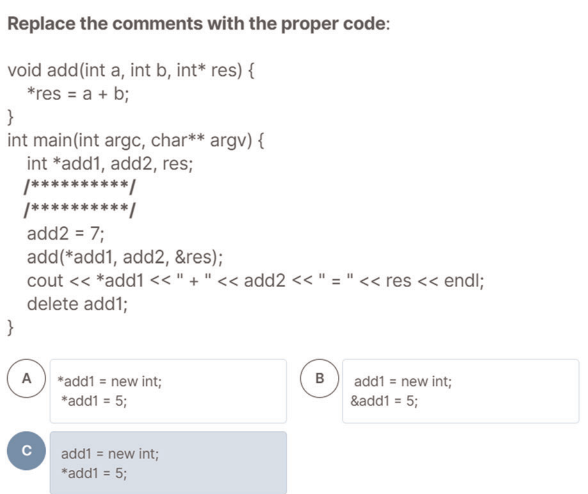

FIGURE 1 An example of a Socrative question that students had to answer interactively during a lecture

question about the use of pointers in C++ with three possible answers as it appeared to students to be answered in Socrative.

In those cases in which the overall percentage of correct answers ranged between 30% and 70%, the students were asked to discuss the question in small groups for a few minutes. In those discussions, the students tried to explain to each other what they had answered and why they thought it was the right answer. Next, the students answered the question again through Socrative based on the new evidence provided by the discussion with their classmates. Finally, the professor revealed the right answer and discussed the question with the students. Questions that were correctly answered in a percentage greater than 70% did not need to be discussed in groups. Those questions were quickly explained by the professor and only required more time if a student asked for more details. On the other hand, questions that obtained a hit ratio lower than 30% were explained directly by the professor in detail, as the concepts related to those questions were assumed to be clearly hard to understand, and therefore a deeper explanation and discussion were required.

Nevertheless, peer instruction was not the only strategy that we used to conduct in-class activities. It is very common [3], and our experience was not an exception, for the students to be very reluctant to ask questions, share their doubts, and participate in discussions even when they had not understood the concepts through out-of-class self-study. In an attempt to avoid this situation, we interleaved questions/discussions with short and incremental programming exercises intended to explain and exemplify the main concepts of the lesson step by step as a minilecture. These incremental exercises served as a common thread between lectures and had a twofold goal: (1) avoid posing different types of exercises in each lecture, requiring no extra time for explaining new practical cases, and (2) allow professors to use these exercises for elaborating the homework questionnaires on them. These exercises were not time-consuming, leaving most of the lecture time for interaction between students. Several recent studies [20] also found that these kinds of short minilectures combined with flipped classroom activities are more deployable in large classes, as was the case with our groups of about 100 students.

At the end of each lecture, the self-check questionnaires answered by students as homework were also revised and briefly discussed to provide the students with convenient feedback. To conduct this discussion, those questions with a low percentage of correct answers were presented to students to be discussed in small groups and answered again by using Socrative. Since that discussion is carried out at the end of the lecture, students have a more strengthened knowledge of the concepts and therefore much better results were usually obtained than in the self-check questionnaire.

3.3 | Statistical analysis

To study differences between grades in the final exam and laboratory exams obtained each academic year, an analysis of covariance (ANCOVA) in IBM SPSS 24 was performed. In this analysis, grades were set as the dependent variables, academic year as the factor, and, to reduce the effect of the personal capacity of each student on the results, the admission grade was set as a covariate. Since several groups (academic years) were compared simultaneously, the ANCOVA analysis was configured using a post-hoc Bonferroni test to correct for multiple comparisons and also to identify the pairs of significantly different groups.

To study the relationship between the homework and the performance of the students, we compute correlations between the number of self-check questionnaires answered, the grade obtained in those questionnaires, and the final exam grade. These correlations were computed in MATLAB 2013a using the command *partialcorr*. This command obtained the sample linear partial correlation coefficients (ρ) between the measures, controlling for the admission grade to reduce the effect of the personal capacity of each student.

Comparisons in the dropout rate and the pass rate (i.e., the percentage of students passing the final exam) between academic years were performed in SPSS 24 using the Pearson χ^2 test, comparing each pair of proportions using Z tests. These χ^2 tests were also configured using a post-hoc Bonferroni test to correct for multiple comparisons (multiple academic years). Analogous χ^2 tests were performed to compare the rates in students' opinion questionnaires.

All results in statistical analyses were considered significant when the p-value obtained was below .05. Box-plots for comparing the distribution of each measure graphically and linear regression graphs for visualizing correlations were computed in MATLAB 2013a.

4 | RESULTS

We obtained and analyzed data from the academic results and rates achieved by the students, data of participation in the activities developed to implement flipped

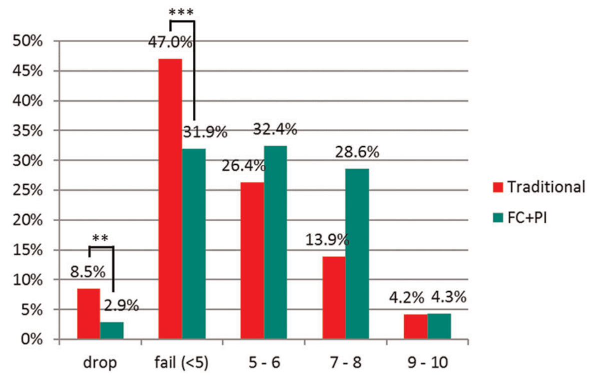

FIGURE 2 Comparison between the dropout rate and the grade distribution from academic years with traditional teaching versus academic years in which flipped classroom and peer instruction (FC + PI) were used. p Values correspond to a Z-test analysis corrected using the Bonferroni method for multiple comparisons. Only p values for the dropout rate and the fail rate are shown ($*p < .05$, $**p < .01$, $***p < .001$)

TABLE 2 Rates and academic results

Year	Lab attendance	Lab exam 1	Lab exam 2	Final exam	Pass rate	Dropout rate
2013/14	70%	4.44 (2.22)	4.27 (2.38)	4.18 (2.57)	45%	5%
2014/15	77%	4.92 (2.51)	6.00 (2.75)	4.62 (2.30)	48%	9%
2015/16	77%	4.99 (2.74)	6.82 (2.74)	3.78 (2.66)	37%	16%
2016/017	76%	5.76 (3.00)	7.27 (2.68)	5.54 (2.72)	61%	4%
2017/18	80%	6.46 (2.92)	7.67 (1.99)	5.94 (2.42)	66%	3%
2018/19	79%	6.57 (3.15)	8.24 (1.72)	5.60 (2.28)	68%	3%
p Value	<.001[a]	<.001[b]	<.001[b]	<.001[b]	<.001[a]	<.001[a]

Note: Lab Attendance: Average percentage of students attending the lab sessions; lab exam 1: Grade of the lab exam in the middle of the semester (out of 10); lab exam 2: Grade of the lab exam at the end of the semester (out of 10); final exam: Grade of the final written theory exam (out of 10); pass rate: Students passing the final exam/students taking the final exam; and dropout rate: Students not taking the final exam/students. Grade values are shown as mean (standard deviation).

[a] p Value for χ^2 test.

[b] p Value for analysis of covariance analysis.

classroom and peer instruction and their impact, and also the students' opinions about the methodology used. All these results are detailed in this section.

4.1 | Academic results

Table 2 shows the academic results and rates obtained in the six academic years that we analyzed. Overall, these results show that all academic results and rates improved for those offerings in which flipped classroom and peer instruction were applied (academic years 2017/18 and 2018/19). Figure 2 shows the percentage distribution of

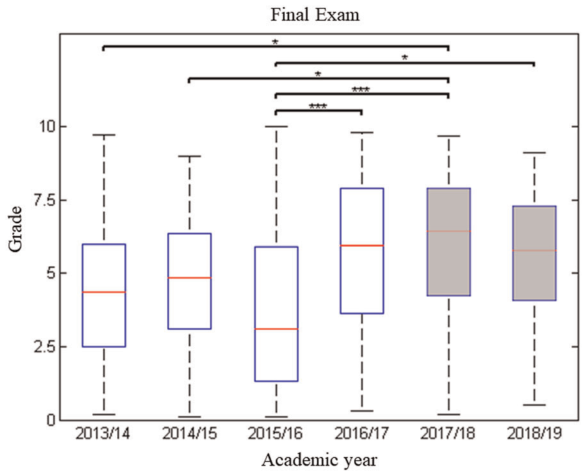

FIGURE 3 Boxplot with differences between academic years in the final exam grade. *p* Values correspond to analysis of covariance analysis (admission grade as covariate) with the post-hoc Bonferroni multiple comparison test. Only *p* values below .05 are displayed (*$p < .05$, **$p < .01$, ***$p < .001$). Shaded boxes show data for academic years in which flipped classroom was applied

the grades for the final exam comparing academic years with traditional teaching to academic years in which flipped classroom was used. These results show that the dropout rate and fail share (i.e., percentage of students not passing the exam) decreased significantly and better grades were also obtained when the new methodology was used.

Figures 3 and 4 show the box plots and significant differences between academic years in the final exam grade and the laboratory exam grades, respectively. Table 2 shows that the average grades for the final exam and laboratory exams were higher in those academic years where flipped classroom was used. Pair comparisons revealed significant differences in the average final exam grades between the academic year 2017/18 and academic years 2013/14, 2014/15, and 2015/16, and between the academic year 2018/19 and the academic year 2015/16 (see Figure 3). The average grade of the final exam for the academic year 2016/17 was also significantly greater than the one for the academic year 2015/16 (see Figure 3). Similarly, the average grade for the laboratory exam 1 was significantly higher in the academic year 2017/18 with respect to academic years 2013/14, 2014/15, and 2015/16 (see Figure 4—lab exam 1). For the case of laboratory exam 2, both academic years 2017/18 and 2018/19 showed average grades significantly higher than academic years 2013/14 and 2014/15 (see Figure 4—lab exam 2).

Table 2 and Figure 2 show that the pass rate results were better in the academic years where flipped classroom was used. We have also compared the average pass rate obtained for each different professor (see Figure 5). These results show that all professors experienced an increase in the pass rate of their students when using flipped classroom. This increase was statistically significant in the case of one particular professor.

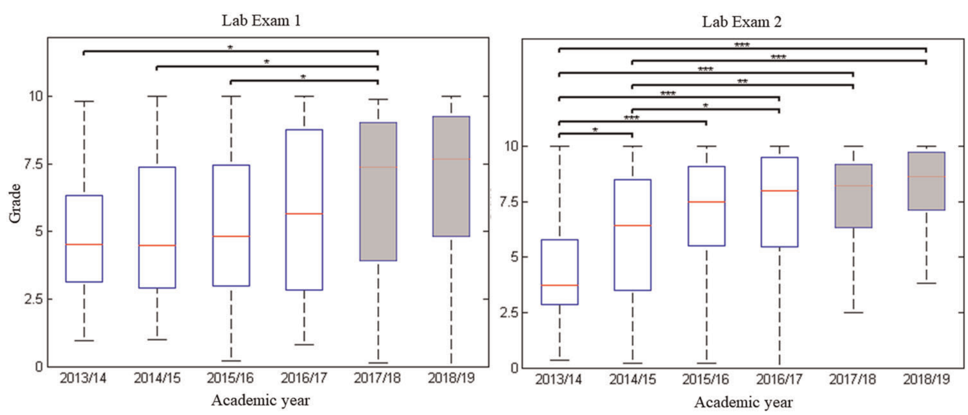

FIGURE 4 Boxplots with differences between academic years in the grade obtained in each laboratory exam. *p* Values correspond to analysis of covariance analysis (admission grade as a covariate) with the post-hoc Bonferroni multiple comparison test. Only *p* values below .05 are displayed (*$p < .05$, **$p < .01$, ***$p < .001$). Shaded boxes show data for academic years in which flipped classroom was applied

4.2 | Impact of flipped classroom activities

To evaluate to which extent out-of-class activities influenced the students' performance, we have analyzed the number of self-study questionnaires answered, the grade achieved in these tests, and their relationships to the final exam grade. This analysis covers academic years 2017/18 and 2018/19, the ones in which flipped classroom and peer instruction were applied. Table 3 shows that no significant differences were found for the aforementioned three measures between the two academic years. The average percentages of the number of questionnaires answered were 71% (course 2017/18) and 77% (course 2018/19). We computed partial correlations (controlling for the admission grade), and the results revealed moderate to strong positive correlations between the number of self-study tests answered and the final exam grade ($\rho = 0.45$, $p < .001$), and between the grade obtained in those tests and the final exam grade ($\rho = 0.51$, $p < .001$). Figure 6 shows these results graphically.

To assess the direct impact of the discussions in which students participate in class as peer instruction activities, we compared the hit rate shown in the Socrative platform for the first and the second time each question was answered. Table 4 summarizes these results. Those questions that required peer discussion were answered again with a hit rate increase of 9% (2017/18) and 13% (2018/19) regarding the first individual answer. Hit rates for the first answer and hit rate increases were found to be statistically similar between academic years ($\chi^2 = 0.66$, $p = .41$ and $\chi^2 = 0.02$, $p = .89$, respectively). However, the hit rate increase after discussion was found to be statistically nonsignificant in each academic year ($\chi^2 = 1.71$, $p = .19$ and $\chi^2 = 3.18$, $p = .07$, respectively).

Laboratory attendance is also a clear indicator of the student commitment to the course, and increasing this commitment was one of the main reasons we implemented the flipped classroom methodology. This rate presented the same trend that we found for the academic results, as Table 2 shows. The lab attendance in the academic years 2017/18 and 2018/19 increased with respect to the academic years in which the traditional teaching methodology was used. The increase in lab attendance when flipped classroom was applied was statistically significant between the academic years 2017/18 and 2013/14 ($\chi^2 = 53.27$, $p < .001$), 2017/18 and 2015/16 ($\chi^2 = 5.39$, $p < .05$), 2017/18 and 2016/17 ($\chi^2 = 10.61$, $p < .01$), 2018/19 and 2013/14 ($\chi^2 = 40.19$, $p < .001$), and 2018/19 and 2016/17 ($\chi^2 = 6.04$, $p < .05$). Overall, a significant increase in lab attendance from 75% to 79% ($\chi^2 = 27.62$, $p < .001$) was obtained when flipped classroom was adopted.

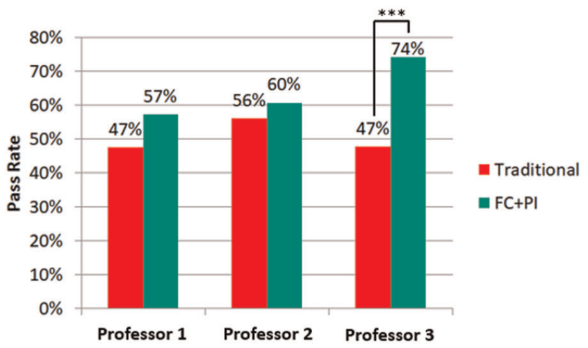

FIGURE 5 Pass rate for professors using both traditional teaching and flipped classroom (FC + PI) in different offerings of the course. *p* Values correspond to *Z*-test analysis corrected with the Bonferroni method for multiple comparisons. Only *p* values below .05 are displayed (*$p < .05$, **$p < .01$, ***$p < .001$)

4.3 | Students' feedback

Students took an anonymous survey at the end of every academic year so that the professors could know the students' opinion about the course. These surveys had two main questions in common for course offerings in both methodologies: (1) "Do you think the dynamics of the theoretical lectures is adequate?" and (2) "Did you work on the lecture topics before attending classes?". Table 5 shows the percentage of students who answered "yes" to these two questions

TABLE 3 Self-study questionnaire results

Year	Number questionnaires	Grade questionnaires	Final exam
2017/18	13.46 (5.30)	6.34 (2.94)	5.94 (2.42)
2018/19	14.66 (5.17)	6.81 (2.88)	5.60 (2.28)
p Value	.254	.436	.484

Note: Number questionnaires: Number of self-study questionnaires answered (out of 19); grade questionnaires: Average grade obtained in the self-study questionnaires answered (out of 10); and Final exam: Grade of the final written theory exam (out of 10). Values are shown as mean (standard deviation). *p*-value for analysis of covariance analysis.

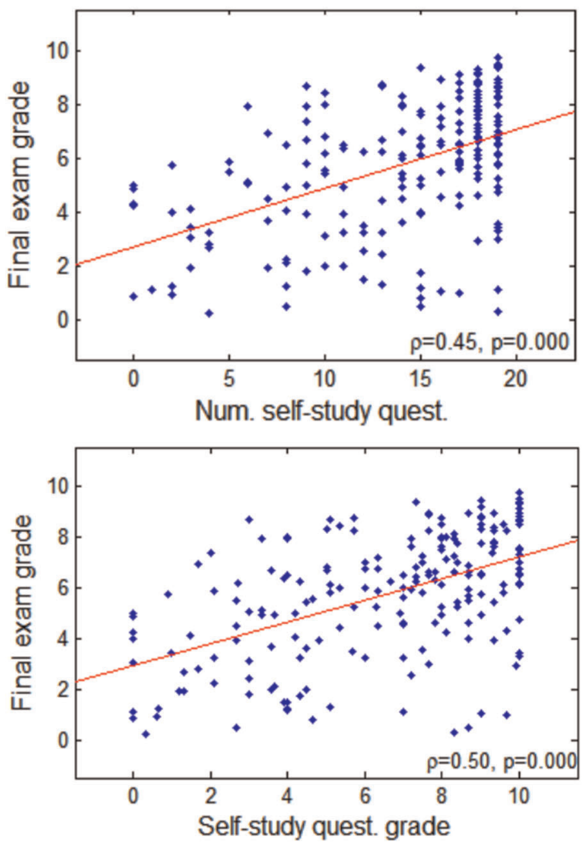

FIGURE 6 Partial correlations between the number of self-study questionnaires answered (top), the grade achieved in these tests (bottom), and the final exam grade

TABLE 4 Hit rate for Socrative questions

Year	First answer	Increase after discussion	*p* Value
2017/18	53%	9%	.19
2018/19	47%	13%	.07
p Value	0.41	0.89	

Note: First answer: Average hit rate for Socrative questions presented for the first time to students; increase after discussion: Average hit rate increase for Socrative questions presented again to students after peer discussion. *p* Value for the χ^2 test.

each academic year. No significant differences were found in the students' opinion about the dynamics of the lectures. Nevertheless, the percentage of students who prepared the lectures in advance was significantly different between academic years. Specifically, this percentage significantly increased from 50% in traditional teaching courses to 88% in those courses where flipped classroom was applied ($\chi^2 = 46.59$, $p < .001$).

In these surveys, we also asked the students for their opinions about the usefulness of the peer instruction methodology (see Table 5). On average, 90% of the students thought that interactions with their classmates provided by the methodology helped them solve their questions (no significant differences were found between the two offerings where peer instruction was used). Finally, one of our main concerns was whether the homework workload needed for flipped classrooms was adjusted to what was planned in each learning card or not. According to the students' answers (see Table 5), on average, 92% of the students needed the same or less time than expected to perform the activities at home (no significant differences were found between the two course offerings with the flipped classroom methodology).

5 | DISCUSSION

Our results, after applying flipped classroom and peer instruction, showed that all academic results and rates improved. These results are in line with those obtained in several previous studies on the application of flipped classroom in introductory courses on computer programming. Jonsson's [12] experiment showed better results in the final exam, although his study was limited to only one academic year. Similar to that study, our results, based on a sample from six academic years, indicate that not only were better pass rates obtained but also the performance of the students improved, as a huge number of students

TABLE 5 Results of students' opinion surveys

Year	*N*	Lecture dynamic is adequate	Student prepare lectures	Peer instruction is useful	Homework time estimation is correct
2015/16	88	86%	47%	–	–
2016/17	53	81%	57%	–	–
2017/18	72	82%	83%	86%	91%
2018/19	60	83%	96%	97%	94%
p Value		0.83	$p < .001$	0.08	0.71

Note: N: number of students who answered the survey; *p* values for the χ^2 test.

managed to obtain higher grades (see Figure 2). Nevertheless, contrary to Jonsson's study, our share of excellent grades (from 9 to 10 out of 10) remained very similar for both teaching methodologies. Our impression is that excellent students obtain excellent marks mainly due to their own capacity and motivation, so they are less affected by the teaching methodology used.

Average rates did not only improve between different offerings of the course but also for students assigned to each professor (see Figure 5). The three professors obtained better results (pass rate) when they used flipped classroom and peer instruction. This fact is very relevant because it proves that the methodology improved students' results regardless of the professor.

Another important consequence of applying flipped classroom was the increase in student engagement in all the course activities. As a result, the dropout rate significantly decreased from 9% to 3% on average (Figure 2). The rates of laboratory attendance also increased significantly (Table 2), with better grades in both laboratory exams when the new methodology was used (Figure 4). These good results overcame known challenges of flipped classroom related to decreased class attendance in large courses [9,5]. Previous studies also showed that peer instruction by itself could not be enough to improve class attendance [16], so we believe that the reason for our good results in laboratory attendance could be the use of both flipped classroom and peer instruction, which improved self-learning and facilitated discussion and participation in the classroom.

We want to highlight the increase in the average grade obtained in the first laboratory exam, the one that takes place in the middle of the semester (Table 2 and Figure 4). We believe that the continuous follow-up of the course has more influence on this kind of exam than the typical extra effort made just before taking any final exam. The students reported this fact in the final course survey, in which they recognized a significantly higher implication in preparing classes when flipped classroom was used (88% compared to 50% for the case of traditional teaching). The higher degree of commitment to the course had a logical impact on concept acquisition by students as revealed by the significant positive correlations found between homework completion and the final exam grade obtained (see Figure 6). These results are in line with recent studies showing the association between students' consistency in performing the out-of-class activities and their learning performance [13].

Demographic data in Table 1 show that the admission grade of the students was higher for the two academic years where flipped classroom was used. To reduce biases in the statistical analyses due to differences in the personal capacity of students between academic years, this admission grade was incorporated as a covariate in all statistical models, thus removing its effects from the statistical tests. Nevertheless, groups with students of a higher level facilitate the teaching process, generating a more adequate learning atmosphere, which usually has a positive influence on those other students who are not so good. This fact cannot explain by itself the results obtained but could affect somehow have a positive effect.

Peer instruction also provided good results, allowing the students to improve their number of right answers after peer discussion by average rates of 9% and 13% for the two offerings of the course in which the methodology was changed. Taking into account that the initial average hit rates when students answered without peer discussion were 53% and 47%, the final average hit rate for Socrative questions was around 60% for both academic years. This rate is slightly lower than but similar to the final pass rate obtained for those two academic years. This would mean that continuous work motivated by flipped classroom activities and discussions offered by peer instruction allowed the students to acquire the knowledge needed to eventually pass the final exam.

Surveys answered by students at the end of the course offerings indicated that flipped classroom had a major impact on the students' homework and class preparation. The average percentage of students who worked on the topics before the lecture increased considerably from 50% to 88%. We believe that this result is due to the use of flipped classroom in the course because adequate preparation before the lectures is essential to optimize the use of time in the classroom. Furthermore, a subject such as computer programming, whose concepts are introduced incrementally and related to one another, especially needs that kind of daily work.

The main concern we had in mind when we decided to implement the new methodology was how students were going to accept the change of roles in the classroom, where they were going to have a much more active role due to peer-instruction activities and extra homework for lecture preparation. Fortunately, according to their answers in the surveys, the new methodology was valued as adequate by most of the students. Also, most of the students stated that they only needed the estimated time or even less for completing the homework. Moreover, they thought that classmate interactions provided by the methodology helped them solve their questions. We believe that these good results in students' opinions could not have been achieved without a careful design of both the out-of-class activities and the in-class Socrative questions.

We experienced some difficulties when implementing the new methodology, as reported by other authors in previous studies [1]. Among them, we want to highlight that it was hard for us to adjust in-class activities to the time available (1 h for each theory session). Two strategies helped us in this issue:

(1) Before each class, we checked the students' answers to the corresponding self-study questionnaires to detect easy concepts that did not need additional explanations. These easy concepts were not included in the examples and exercises proposed in the classroom.
(2) In the lectures, we provided only brief overviews of those concepts for which Socrative questions were correctly answered at high rates. For example, if a Socrative question obtained 90% of correct answers the first time it was proposed, then a very brief explanation (1 min or less) of the correct answer was provided and, obviously, this question was not proposed for peer discussion.

6 | LIMITATIONS

This study has analyzed six independent and consecutive offerings of the same course, so no experimental controls were imposed. Nevertheless, all six offerings maintained the same design, the same topics, and, importantly for this study, the same structure for both laboratory and final exams.

Our resources for the implementation of the flipped classroom did not include video lectures. Although that resource is usually used [1], several authors have also claimed that it may be better to focus on carefully selecting in-class instruction methods rather than devoting considerable time and resources to developing online videos [15]. Our starting point was the set of carefully designed resources already used with traditional teaching, including detailed lecture slides, many code examples, and readings from recognized textbooks. We also elaborated new material to guide out-of-class activities, including exercises and questionnaires. Nevertheless, we are already working on including mini-videos to explain selected key concepts in the next offerings of the course. Further analysis will be needed to assess the impact of this new resource on the students' performance.

7 | CONCLUSIONS

We presented the results obtained when using flipped classroom and peer instruction in an introductory course of programming. These results suggest that these methodologies contributed toward improving students' engagement, class attendance, and participation, and as a result, the overall students' performance. Significantly better pass ($\chi^2 = 12.94$, $p < .001$) and dropout rates ($\chi^2 = 7.08$, $p < .01$) were obtained when using flipped classroom and peer instruction. These methodologies not only improved the average rates between different offerings of the course but also for students assigned to each professor. This fact proves that these two methodologies improved students' results regardless of the professor. Laboratory attendance also improved significantly ($\chi^2 = 27.62$, $p < .001$), and significant positive correlations between the work performed at home and the grade obtained in the final exam were obtained ($\rho = 0.51$, $p < .001$). The average percentage of students who prepared the lecture in advance increased significantly from 50% to 88% ($\chi^2 = 46.59$, $p < .001$). This last result is crucial because an adequate preparation before the lectures is essential to optimize the use of time in the classroom. Finally, we want to remark that this improvement in the academic results and the student commitment was noticeable not only in the figures but also in the overall perception of the students.

ACKNOWLEDGMENTS

This study was partially supported by the University of Jaén through the grant number PID22_201617.

ORCID

Juan Ruiz de Miras http://orcid.org/0000-0001-7579-8350

REFERENCES

1. G. Akcayir and M. Akçayır, *The flipped classroom: A review of its advantages and challenges*, Comput. Educ. **126** (2018), 334–345. https://doi.org/10.1016/j.compedu.2018.07.021
2. S. D. Anastasiadou and A. S. Karakos, *The beliefs of Electrical and Computer engineering students' regarding computer programming*, Int. J. Technol. Knowl. Soc. **7** (2011), 37–51. https://doi.org/10.18848/1832-3669/cgp/v07i01/56170
3. D. Baldwin, "*Can we 'flip' non-major programming courses yet?*," SIGCSE 2015—Proc. 46th ACM Tech. Symp. Comput. Sci. Educ., 2015, pp. 563–568. https://doi.org/10.1145/2676723.2677271
4. T. Beaubouef, and J. Mason, *Why the high attrition rate for computer science students: Some thoughts and observations*, ACM SGCSE Bull. **37** (2005), no. 2, 103–106.
5. J. Campbell, D. Horton, M. Craig, and P. Gries, "*Evaluating an inverted CS1*," SIGCSE 2014—Proc. 45th ACM Tech. Symp. Comput. Sci. Educ., 2014, pp. 307–312. https://doi.org/10.1145/2538862.2538943
6. C. H. Crouch and E. Mazur, *Peer instruction: Ten years of experience and results*, Am. J. Phys. **69** (2001), 970–977. https://doi.org/10.1119/1.1374249

7. S. J. DeLozier and M. G. Rhodes, *Flipped classrooms: A review of key ideas and recommendations for practice*, Educ. Psychol. Rev. **29** (2017), 141–151. https://doi.org/10.1007/s10648-015-9356-9
8. S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, *Active learning increases student performance in science, engineering, and mathematics*, Proc. Natl. Acad. Sci. U. S. A. **111** (2014), 8410–8415. https://doi.org/10.1073/pnas.1319030111
9. M. N. Giannakos, J. Krogstie, and N. Chrisochoides"*Reviewing the flipped classroom research: Reflections for computer science education*," Proc. CSERC 2014: Comput. Sci. Educ. Res. Conf., 2014, pp. 1–7. https://doi.org/10.1145/2691352.2691354
10. D. Horton, M. Craig, J. Campbell, P. Gries, and D. Zingaro, "*Comparing outcomes in inverted and traditional CS1*," ITICSE 2014—Proc. 2014 Innov. Technol. Comput. Sci. Educ. Conf., 2014, pp. 261–266. https://doi.org/10.1145/2591708.2591752
11. L. Hurtubise, E. Hall, L. Sheridan, and H. Han, *The flipped classroom in medical education: Engaging students to build competency*, J. Med. Educ. Curric. Dev. **2** (2015), 35–43. https://doi.org/10.4137/jmecd.s23895
12. H. Jonsson, "*Using flipped classroom, peer discussion, and just-in-time teaching to increase learning in a programming course*," Proc. Front. Educ. Conf., FIE, 2015, pp. 1–9. https://doi.org/10.1109/FIE.2015.7344221
13. J. Jovanovic, N. Mirriahi, D. Gašević, S. Dawson, and A. Pardo, *Predictive power of regularity of pre-class activities in a flipped classroom*, Comput. Educ. **134** (2019), 156–168. https://doi.org/10.1016/j.compedu.2019.02.011
14. Kanika, S. Chakraverty, and P. Chakraborty, *Tools and techniques for teaching computer programming: A review*, J. Educ. Technol. Syst. **49** (2020), no. 2, 170–198. https://doi.org/10.1177/0047239520926971
15. R. H. Kay and T. Macdonald, "*Comparing flipped, active, and lecture-based teaching approaches in higher education*," 2016. https://doi.org/10.13140/RG.2.1.4727.4481
16. C. M. Keet, "*An experiment with peer instruction in computer science to enhance class attendance*," 23rd Annu. Meeting Southern African Association for Res. Math., Sci., and Technol. Educ., 2014, pp. 319–331.
17. C. B. Lee, "*Experience report: CS1 in MATLAB for non-majors, with media computation and peer instruction*," SIGCSE 2013—Proc. 44th ACM Tech. Symp. Comput. Sci. Educ., 2013, pp. 35–40.
18. M. Lepp and E. Tonisson, "*Integrating flipped classroom approach and work in pairs into workshops in programming course*," Int. Conf. Front. Educ.: Comput. Sci. Comput. Eng., 2015, pp. 220–226.
19. M. Lou Maher, C. Latulipe, H. Lipford, and A. Rorrer, "*Flipped classroom strategies for CS education*," SIGCSE 2015—Proc. 46th ACM Tech. Symp. Comput. Sci. Educ., 2015, pp. 218–223. https://doi.org/10.1145/2676723.2677252
20. S. Mader and F. Bry, "*Phased classroom instruction: A case study on teaching programming languages*," in CSEDU 2019—Proc. 11th Int. Conf. Comput. Supported Educ., 2019, pp. 1–11. https://doi.org/10.5220/0007655702410251
21. M. Marqués, "*What's behind flipped classroom*," XXII Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI), 2016, pp. 77–84.
22. E. Mazur, Peer instruction: A user's manual, Prentice Hall, Upper Saddle River, NJ, 1997.
23. L. R. Murillo-Zamorano, J. Á. López Sánchez, and A. L. Godoy-Caballero, *How the flipped classroom affects knowledge, skills, and engagement in higher education: Effects on students' satisfaction*, Comput. Educ. **141** (2019), 1–18. https://doi.org/10.1016/j.compedu.2019.103608
24. G. M. Novak, Just-in-time teaching: Blending active learning with web technology, Prentice Hall, Upper Saddle River, NJ, 1999.
25. A. Oram, and G. Wilson, Making software—what really works, and why we believe It, O'Reilly Media, UK, 2010.
26. H. Özyurt and Ö. Özyurt, *Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy,"*, Comput. Appl. Eng. Educ. **26** (2018), 2036–2046. https://doi.org/10.1002/cae.21973
27. L. Porter, C. Bailey-Lee, and B. Simon, "*Halving fail rates using peer instruction: A study of four computer science courses*," SIGCSE 2013— Proc. 44th ACM Tech. Symp. Comput. Sci. Educ., 2013, pp. 177–182.
28. A. Sams, and J. Bergmann, Flip your classroom: Reach every student in every class every day, International Society for Technology in Education, USA, 2012.
29. N. Sarawagi, *A flipped CS0 classroom: Applying Bloom's taxonomy to algorithmic thinking*, J. Comput. Sci. Coll. **29** (2014), no. 6, 21–28.

AUTHOR BIOGRAPHIES

Juan R. de Miras is a senior lecturer at the Software Engineering Department of the University of Granada (Spain). He received an MS degree in Computer Science in 1995 and a PhD in Computer Science in 2001. His main research interests are in Computer Graphics, Geometric Modelling, High-Performance Computing, Computer Applications in Biomedicine, and E-learning.

José-Ramón B. Almagro graduated in Computer Science from the University of Granada in 1994. In 1996, he joined the Department of Computer Science at the University of Jaén as a lecturer, where he is still working today. Between 2007 and 2015, he was Director of E-learning and On-line Services at the University of Jaén. His research interests include Web Application Development, Cloud Computing, Cybersecurity, and E-learning.

Ángel L. García-Fernández received his BSc degree in Business Computer Science in 1998 from the University of Jaén, and his MSc degree in Computer Science from the University of Granada in 2000. In 2007, he completed his PhD from the University of Granada. He was a research assistant at the University of Jaén from 2001 to 2004, and since 2004, he is a lecturer at the Computer Science Department of the same university. His research interests include Computer Graphics, Computational Archaeology, and Smart Environments.

How to cite this article: J. Ruiz de Miras, J. R. Balsas-Almagro, and Á. L. García-Fernández, *Using flipped classroom and peer instruction methodologies to improve introductory computer programming courses*, Comput. Appl. Eng. Educ. (2021), 1–13. https://doi.org/10.1002/cae.22447