Research to Practice in Computer Programming Course using Flipped Classroom

1st Liang Zhang
School of Computer Science and Engineering
Beihang University
Beijing, China
liang.z@buaa.edu.cn

2nd Jianwei Niu
School of Computer Science and Engineering
Beihang University
Beijing, China
niujianwei@buaa.edu.cn

***Abstract*—This Research to Practice Full Paper presented a Flipped Classroom (FC) approach to teaching a computer programming course. This approach increased students' academic performance, course satisfaction and learning motivation by providing more time for active learning.**

There is an intense need for the studies of FC approach in higher education, especially in computer programming courses. Although FC pedagogy has achieved great success in K-12 schools, there is rare quantitative research on programming courses in higher education.

This paper introduced a FC approach to teaching computer program course, and described in detail how to arrange pre-class activities and in-class activities. Pre-class activities were recommended to be arranged by time. 5 days before class, release course material and open access to programming practice. Students began to study and practice by themselves. 2 days before class, Students were required to complete pre-class test. 1 day before class, the teaching assistant provided a pre-class test analysis report. The lecturer prepared the coming class based on the report. In-class activities consisted of four parts. First, the lecturer explained in detail the common problems in the pre-class test. Second, the lecturer checked the problem-solving by asking questions. Third, students discussed in teams. Fourth, students gave presentations on the advanced topics.

To assess the effectiveness of the FC approach, the study was conducted at the course Swift Language Programming Practice for undergraduate students. Students were divided into two groups. One group implemented traditional pedagogy, and the other group implemented FC pedagogy. The effect of the two groups was analyzed by teaching data collected during 2019-2021. The nonparametric independent-samples Kruskal–Wallis test was used to measure the changes in course performance and student satisfaction. The results showed that by introducing FC to teach computer programming courses, course performance (measured by examination results) and student satisfaction (measured by course questionnaires) were significantly improved.

The contribution of this paper was to propose a FC approach suitable for college programming course, and through quantitative statistical methods to analyze the teaching effect of the course. The results showed that the approach significantly improved the course performance and student satisfaction. The FC approach of this paper can be regarded as a reference for similar course in college.

***Index Terms*—active learning, blended learning, flipped classroom, higher education, computer programming course**

Jianwei Niu is the corresponding author(email:niujianwei@buaa.edu.cn).

I. INTRODUCTION

Major construction in accordance with the international engineering education professional certification system is the only way for the development of China's engineering major. One way of achieving it is to flip the classroom. A Flipped classroom (FC) refers to changing the learning materials originally completed in class to be completed at home and completing homework in class [1]. FC is an effective pedagogy to implement active learning. According to Bonwell and Eison, teachers who want to promote active learning should focus less on transmitting information to students and more on developing students' skills [2]. Students should be encouraged to pursue "higher-order thinking", such as "analysis, synthesis, and evaluation" of information, and to "explore their own attitudes and values" through such activities as writing and discussion [2]. FC is a promising approach to free class time for more active learning [3]. FC has become a popular new instructional model [4], [5]. As FC is mostly practiced in K-12, many flipped learning models are being developed and debates on their pedagogical values continue [6]. What about the use of FC in higher education? In colleges, FC mainly refers to the completion of basic knowledge learning through pre-class preview, while more active learning can be carried out in the classroom [7]. The application of FC in college engineering education is conducive to promoting students' active thinking and communication [7]. Although FC teaching method has been widely used in higher education, evidence-based best practice of FC is still lacking [8], [9].

This study was to propose a FC approach suitable for computer programming courses in higher education. This FC method and traditional method were applied to teaching computer programming course in parallel. To compare the differences between flipped class and traditional class, statistical methods were used to measure the changes in academic performance and course evaluations from 2019 to 2021.

The remainder of this paper is structured as follows. Section II presents related work from the computer programming education domain and beyond. Section III introduces an overview of the course. Section IV provides a detailed statistical analysis of both the academic performance and the course evaluations. Finally, results are discussed in Section V and conclusions are given in Section VI.

II. Related Work

Numerous studies in engineering education and higher education reported on the effectiveness of FC and Active Learning. With the wide application of FC in college teaching, this pedagogy has received more and more attention. [7]. The use of FC in colleges provides students with more opportunities for active learning. For example, students can choose which materials to learn and whether to repeat them [10]. Research shows that many students feel that FC pedagogy helps them deepen their understanding of learning materials [11]. The reason is that FC provides them with more opportunities to interact with teachers and classmates [12]. Although some students have not adapted to FC, their learning performance has been significantly improved [13].

There are increasing practices and researches on flipping computer programming course.

Vojinovic et al. [14] designed a layered teaching model of laboratory programming based on FC teaching method. The FC model was performed on an assembly language programming subject over six years. The results of the study show that students of FC model groups performed better in terms of motivation and performance and most students showed a positive attitude towards the model.

Kay et al. [15] compared lecture-based, active learning, and FC teaching approaches used in a first-year community college computer programming course. Compared with traditional teaching methods, students are more satisfied with FC pedagogy. However, there is no significant difference between the two methods in improving academic performance.

Chis et al. [16] mentioned that the pedagogy of combining FC with problem-based learning (PBL) is very effective. It can help to improve students' acquire more knowledge and pleasant learning experience.

Lai et al. [17] proposed a FC model called dt-cdio, and studied the impact of streaming experience and computing thinking on FC.

Duruk [18] used statistical methods to analyze and study the impact of FC on the improvement of students' programming ability from three aspects: cognitive flexibility, problem-solving skills and flipped learning readiness (FLR).

The above work applied FC pedagogy to computer programming courses. Most experimental results show that FC pedagogy gained a positive effect on students' performance and satisfaction of courses. It is also conducive to improving students' active learning time. However, the actual teaching effect of FC pedagogy is closely related to the design of specific courses (including the teaching design of courses), the situation of students (students' ability, motivation, learning engagement), etc.

This study reported on the results of a three-year practice in computer programming course in which a traditional lecture course (TC) and a FC were used in parallel. The aim of the work was to compare the effects of TC and FC with rigorous statistical methods by measuring changes in both academic performance and course evaluation.

III. Course Implementation

This FC practice was implemented in the undergraduate course Swift Language Programming Practice (SLPP). The course is a professional course offered in the school of computer science of Beihang University. 80% of the students selected were from the School of Computer Science, and 20% were from other schools of information related majors such as Beijing School, School of Higher Engineering, School of Artificial Intelligence and School of Network Security. As the course was a professional elective course, the number of students choosing courses fluctuated every year, but they all exceeded 40, ensuring the necessary number of samples. Students who chose the course were randomly assigned to TC group (TCG) and FC group (FCG). The number of students in the two groups was roughly the same. The TCG and FCG were taught in parallel from 2019 to 2021. We collected the curriculum teaching data from 2019 to 2021. The teaching team was basically stable in the past three years. It consisted of a lecturer and a teaching assistant. The lecturer was fixed. The teaching assistant was graduate students majoring in computer science and changed once a year.

The SLPP course was worth 1.5 credits. It had a total of 16 double classes (two 45-min sessions with a 5-min break), once a week. The course was held every autumn semester and lasted for 4 months. It taught the basics of Swift Language Programming, including grammar, programming skills, underlying implementation. On this basis, it also taught relevant technologies and engineering development methods of iOS App. In the first three months, the course focused on knowledge teaching, and in the last month, it focused on completing three increasingly difficult engineering projects. The demands of the first two projects were relatively fixed, requiring students to complete them independently. The third project was independently designed by students, with self-customized requirements. The project can be completed by teams. Each student's grade was given according to his or her contribution to the team.

A. Traditional Course Group

From 2019 to 2021, the number of students assigned to TCG was 45, 21 and 21 respectively. The course in TC was implemented once a week, which consisted of two 45-min lectures. In class, Swift Language and iOS app development were taught by slides. In the lecture, there was a question session to ask questions about the main points being explained or the contents taught before. Starting from these questions, the teacher guided students to have an extended discussion. Most of the questions were common, which was also the weakest part of students' understanding and mastery in previous years. These questions helped most students deeply grasp the difficulties, but they were targeted for individual students. Some quizzes were randomly arranged in the classroom to evaluate the phased learning effect of students, and the scores were included in the course performance. After class, the slides were published online for students to review. Slides covered the main contents explained in the course. In addition, in the first lesson, we provided two electronic reference textbooks, which introduced Swift language programming and iOS development technology respectively. The lecture only involved the important and difficult contents in the textbook and students learned the basic knowledge of the textbook by themselves after class.

B. Flipped Course Group

From 2019 to 2021, the number of students assigned to FCG was 44, 20 and 21 respectively. The lecturers and teaching assistants of FCG and TCG were the same. FCG also used the same teaching materials as TCG, including slides and 2 textbooks. FC freed up a lot of active learning time for the course, so FCG provided more extended materials and more challenging project assignments.

Pre-class Activities: 5 days before class, released its topics and outline, slides, video material and opened access to programming practice through the course online platform. Students completed programming exercises on related topics before class. Online Judge system recorded the running results of the program and give feedback to students, but the results wasn't included in the course performance. 2 days before class, online tests were opened for students. Students completed the pre-class test on the same day. Each student had only one chance to test. Through the pre-class test, students knew their weak parts after self-study and could prepare for classroom discussion well. The results of pre-class test were included in the course performance, but its proportion was relatively small. As the test results were part of the course performance, students would strive to achieve good results in the test. Meanwhile the pre-class test score only accounted for a small proportion, so as to avoid students taking the exam with the help of others in order to obtain high scores. One day before class, the teaching assistant analyzed the overall situation of the pre-class test and the common problems in students' knowledge mastery according to the system records, wrote the pre-class test analysis report and submitted it to the lecturer. According to the pre-class test analysis report, the lecturer could prepare class with emphasis on knowledge points not mastered, error-prone questions and discussion topics with appropriate difficulty.

In-Class Activities: They consisted of four parts. The first part lasted about 30 minutes. The lecturer explained in detail the common problems in the pre-class test. The second part lasted about 15 minutes. The lecturer asked some questions to check whether the students who performed poorly in the pre-class test had solved their problems, so as to judge whether it was necessary to arrange teaching assistants to tutor these students separately. The third part lasted about 25 minutes. First of all, the lecturer put forward extensible questions for knowledge points of this class. Then, students discussed in teams. The lecturer and teaching assistants randomly participated in some team discussions. Finally, the student representatives of 1-2 groups explained their results of the group discussion, and the lecturer made a remark. The fourth part lasted about 20 minutes. 2-3 students gave presentations on the advanced topics prepared in advance before class. All students evaluated the quality of the presentations by anonymous scoring. The score was included in the course performance.

The course content of SLPP was arranged as follows.

Class 1: Introduce the course requirements of FC, how to complete specific learning tasks in stages, how to access online judge platform, how to obtain relevant video materials and official documents, and join the course WeChat group to gain instant help. Introduce the preliminary knowledge of the course, including how to program in playground and build an iOS project in Xcode.

Class 2,3: Discuss the basic syntax topics of Swift Language, including basic data types, operators, strings, collection types, control flows, functions and closures.

Class 4-8: Discuss the object-oriented topics of Swift Language, including enumeration types, structures and classes, properties, methods, inheritance, constructors and destructors, error handling, optional chains, type operators, extension mechanisms, protocols, memory management, operator overloading, and generics.

Class 9-12: Discuss the technical topics of iOS App development, including: Foundation Framework, UINavigation Controller, UITableView, UIView, Multi Touches, Gesture Recognition, Data Persistence, NSURLConnection, AsiHttpRequest, AutoLayout and Size Classes.

Class 13-15: Assign three iOS App projects with different difficulty as homework. In class, the lecturer and teaching assistant participated in the discussion of each team, gave necessary help to smooth the process of project development, and provided improvement suggestions.

Class 16: Each team introduced their project design through slides and displayed the effect of the work through iPhone. The judges were composed of the lecturer, teaching assistants and designers of outstanding works in previous years. The judges evaluated the works from the aspects of innovation and creativity, technical difficulty, project development workload, operation effect and technical documents of the project.

IV. Evaluation

According to the teaching data from 2019 to 2021, the FC approach was evaluated from two aspects: course performance and student satisfaction.

A. Course Performance Analysis

The grade of the course adopted the hundred-mark system. It consisted of five parts: pre-class test (20%), presentation (10%), assignments after class (10%), final exam (20%) and engineering project (40%). The pre-class quiz consisted of 12 online quizzes. The test questions were mainly multiple-choice questions, judgment questions and programming questions, which were required to be completed within 1 hour. Each student was required to make a presentation. Each class provided two students with the opportunity to make a report.

The topic of the report was selected from the relevant scope specified by the lecturer. The order of making reports was applied by students on their own initiative, on a first come, first served basis. There were 12 assignments after class, and each assignment accounted for the same proportion of scores. Assignments after class provided a lot of programming exercises for the topics in class. Final exam was dominated by programming questions for a total of 2 hours.

Engineering projects accounted for the largest proportion of course scores. Three projects were required to be completed in this part. The first project was to develop a horizontal version of the scientific calculator. The second project was to develop iOS App 2048. The third project was to design and implement a practical iOS App by teams, which could be practical tools, mobile office, life services, information portal, mobile games and so on. Before class, the lecturer provided relevant materials for engineering development (design patterns document, UML specifications manual, cocoa touch reference manual, etc.). Students learned relevant theoretical knowledge, development technology and the use of relevant tools by themselves. In class, the lecturer introduced the key points and difficulties in the development of the project, participated in the discussion of each team with the teaching assistant, and helped students solve the obstacles encountered in the process of project demand analysis and system design. After class, students independently completed architecture design, system detailed design, coding, system unit test and function test, and App release and launch. After completing the project, students were required to submit project related documents and source codes. The teaching assistant reviewed these files online. The lecturer made a preliminary review of the project offline. Finally, each team demonstrated their own project in class and evaluated each other.

The distribution of scores from each year is shown in table I. The teacher wrote the examination for 2019. The teaching assistants in 2020 and 2021 participated in the preparation of examination in that year respectively. The difficulty and the scope of examination in the three years remained at a roughly equal level.

To compare the difference of the course performance between TCG and FCG from 2019 to 2021, the data of scores are divided into six groups. They are TCG2019, FCG2019, TCG2020, FCG2020, TCG2021, FCG2021. The box plot of the six groups of data is shown in Fig.1.

We used the nonparametric independent-samples Kruskal–Wallis test, which was based on scores and was an overall test of any differences across the six groups. The results are: test statistic = 48.3876, p(2-sided) = 0.000, N = 172, meaning that there are differences between groups, but to know between which groups, post-hoc pairwise comparison tests are needed. There are significant pairwise comparisons ($p<0.001$) for two comparisons, namely, between TCG2019 and FCG2019, TCG2019 and FCG2021, TCG2020 and FCG2019, TCG2020 and FCG2021, TCG2021 and FCG2019, TCG2021 and FCG2021, FCG2019 and FCG2020, FCG2019 and FCG2021, and FCG2020 and FCG2021.

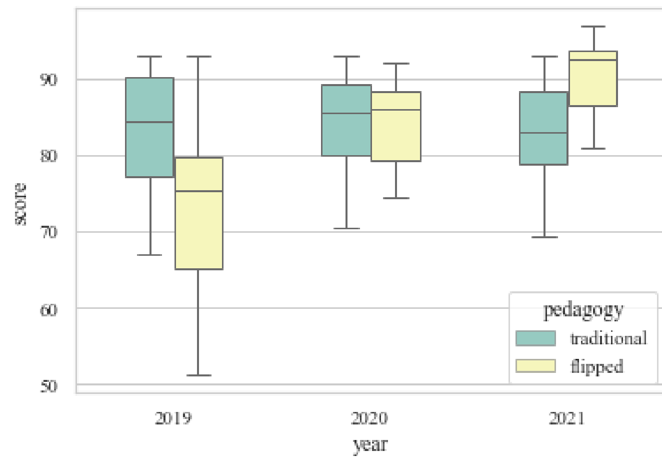

Fig. 1. Comparison of Scores between TCG and FCG 2019-2021.

The nonsignificant comparisons are between TCG2019 and FCG2020 (p = 0.5784), TCG2019 and TCG2020 (p = 0.6043), TCG2019 and TCG2021 (p = 0.8823), TCG2020 and FCG2020 (p = 0.5686), TCG2020 and TCG2021 (p = 0.3865), and TCG2021 and FCG2020 (p = 0.3704).

The result means that the effect of FC pedagogy is obviously not as good as TC pedagogy in the first year (2019). The average score, the lowest score and the middle 50% of students in TCG are higher than those in FCG. The main reasons are that the teaching team lacked the experience of FC pedagogy, the lesson preparation was hasty, the arrangement of course content was not reasonable, and the pre-class learning resources provided were also not rich. In addition, students were exposed to FC pedagogy for the first time, so they inevitably felt at a loss and couldn't cooperate well. There is no significant difference between TCG and FCG in scores in the second year (2020). Compared with 2019, score distribution of TCG does not change much, while FCG's score is significantly improved. Due to the one-year course construction, a lot of convenient teaching material were developed, which was conducive to students' active learning before class. The teaching team also accumulated one year's experience in the implementation of FC pedagogy and iteratively improved the teaching process. Most students had a certain understanding of FC pedagogy and improved the learning effect of FC through active learning before class. In the third year (2021), FCG's scores began to surpass TCG's, especially the average and minimum scores. In the three years, the scores of TCG was relatively stable, which was closely related to the relatively mature and fixed TC pedagogy. The score of FCG was greatly improved, especially from 2019 to 2020, because the implementation of FC pedagogy was still in the stage of continuous development and improvement. From 2020 to 2021, although the score of FCG was improved, the improvement range was significantly narrowed, which was due to the implementation of FC pedagogy and the objective conditions of relevant courses tended to be stable.

TABLE I
DISTRIBUTION OF SCORES 2019-2021

Score	**2019**		**2020**		**2021**		**Grand Total**
	# of students(percentage)		**# of students(percentage)**		**# of students(percentage)**		
	traditional	flipped	traditional	flipped	traditional	flipped	
90-100	12(27%)	4(9%)	4(19%)	3(15%)	3(14%)	13(62%)	39
80-89	17(38%)	7(16%)	11(52%)	10(50%)	12(57%)	8(38%)	65
70-79	15(33%)	15(34%)	6(29%)	7(35%)	5(24%)	0	48
60-69	1(2%)	9(20%)	0	0	1(5%)	0	11
Fail	0	9(20%)	0	0	0	0	9
Year Total	**45(100%)**	**44(100%)**	**21(100%)**	**20(100%)**	**21(100%)**	**21(100%)**	**172**

B. Student Satisfaction Analysis

After each course, the student satisfaction questionnaires were distributed to all students. Through this questionnaire, the teaching effect of the course can be fed back from the perspective of students' feelings. From 2019 to 2021, questionnaires were distributed to TCG and FCG students respectively. The number of students answering each survey and their corresponding response rates are in 2019 (TCG: N=45, response rate 40%; FCG: N=44, response rate 42%), 2020 (TCG: N=21, response rate 50%; FCG: N=20, response rate 48%), 2021 (TCG: N=21, response rate 47%; FCG: N=21, response rate 49%).

The questions related to the evaluation of FC teaching effect in the questionnaire can be divided into two categories. The first kind of questions are related to the overall evaluation of the curriculum rather than the specific pedagogy. The second kind of questions are about the specific implementation effect of FC pedagogy. These questions only appeared in the questionnaire of FCG.

A typical question in the first kind was "What is your overall impression of the course? Rated from 1 (very poor) to 5 (excellent)."

The descriptive statistics for the above question are shown in Fig.2, which shows students' overall impression of the course. The nonparametric independent-samples Kruskal-Wallis test was used again, which was based on grades and was an overall test of any differences across the six groups. The results are: test statistic = 19.4547, p(2-sided) = 0.001, N = 172, meaning that there are differences between groups, but to know between which groups, post-hoc pairwise comparison tests are needed. There are significant pairwise comparisons ($p<0.05$) for two comparisons, namely, between TCG2019 and FCG2019 (p=0.0067), and FCG2019 and FCG2021 (p=0.0003).

When the FC pedagogy was implemented in the first year (2019), the lecturer had no relevant experience and teaching materials were very limited. At the same time, students were not familiar with the new learning process and didn't have any psychological preparation. Therefore, the overall teaching effect was poor. Compared with TC pedagogy, students' overall impression of FC was obviously lower than that of TC. After two years of practice, the teaching team had accumulated a lot of teaching experience and continuously optimized the organization of the teaching process. Online teaching materials had also been continuously improved, which was more conducive to students' active learning. Students gradually adapted to the FC pedagogy. The students' enthusiasm for learning became stronger and stronger. With the increase of students' active learning, the learning effect became better and better. This circular positive incentive process not only led to a significant improvement in students' performance, but also a significant increase in students' overall satisfaction with the curriculum.

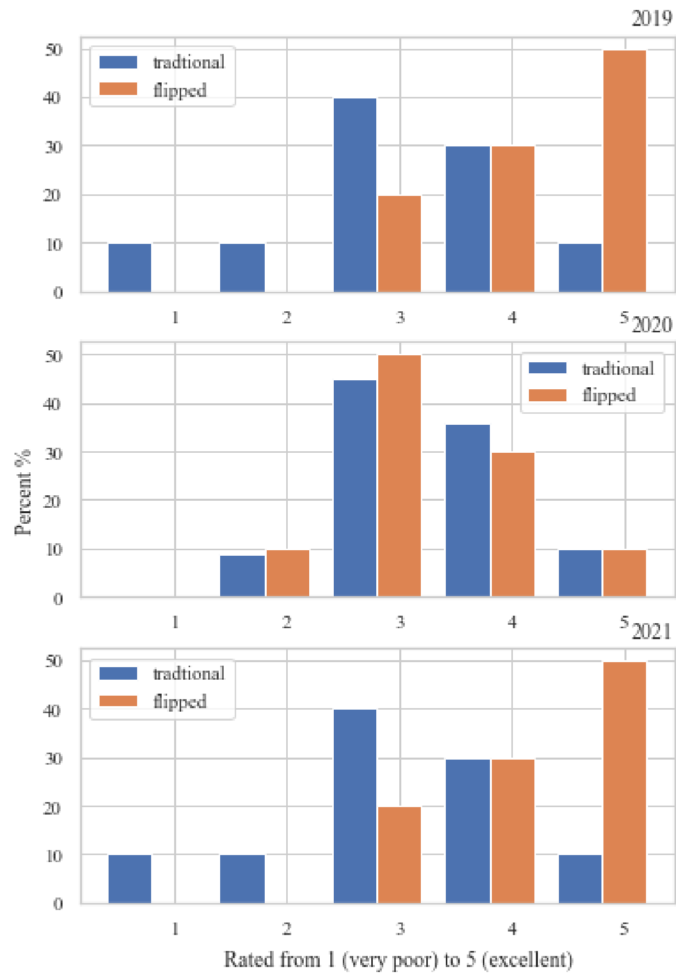

Fig. 2. What is your overall impression of the course? Rated from 1 (very poor) to 5 (excellent).

A typical question in the second kind was "The learning before class was smooth. Rated from 1 (disagree completely) to 5 (agree completely).”

The descriptive statistics for the above question are shown in Fig.3, which shows students’ pre-class learning condition. The nonparametric independent-samples Kruskal-Wallis test was also used, which was based on grades and was an overall test of any differences across the three FCG. The results are: test statistic = 19.3280, p(2-sided) = 0.000, N = 85, meaning that there are significant differences between groups, but to know between which groups, post-hoc pairwise comparison tests are needed. There are significant pairwise comparisons ($p<0.05$) for two comparisons, namely, between FCG2019 and FCG2021 (p=0.000), and FCG2020 and FCG2021 (p=0.0129). The nonsignificant comparisons are between FCG2019 and FCG2020 (p = 0.0902).

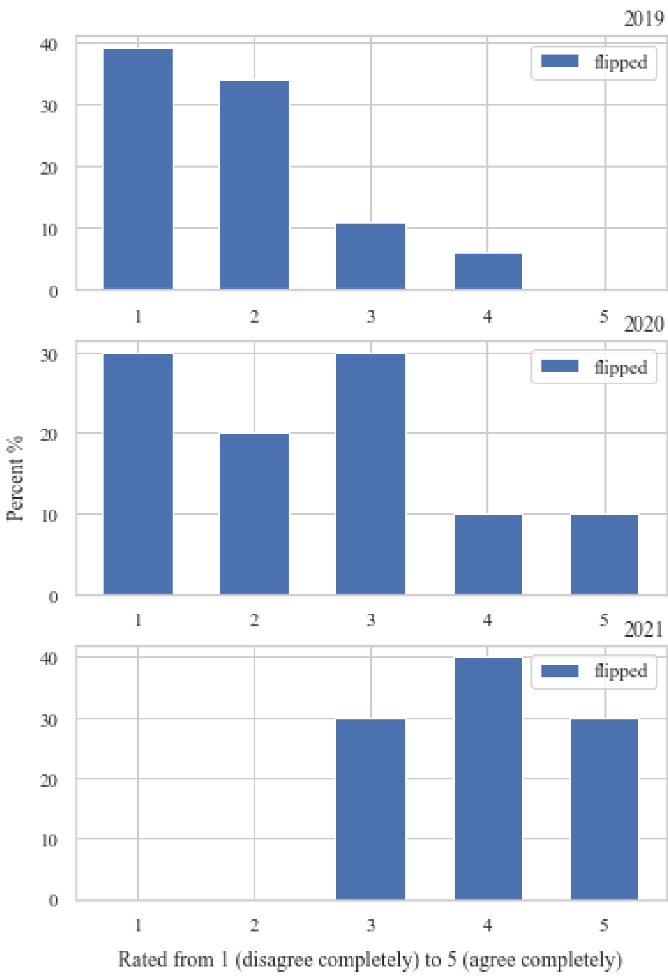

Fig. 3. The learning before class was smooth. Rated from 1 (disagree completely) to 5 (agree completely).

The result means that there was significant promotion in grades the third year of the FC approach, so significant difference between FCG2019 and FCG2021, and FCG2020 and FCG2021. As mentioned earlier, the difficulties of pre-class learning might come from many aspects in 2019. Although the difficulties had been alleviated in 2020, the result of the questionnaires shows that there was still a lot of space for improvement in pre-class learning. In 2021, while the learning effect before class was greatly improved, the overall scores of students were also improved significantly. This shows that pre-class learning is a crucial part of FC.

V. Discussion

Using FC in computer programming course is promising. The results show that after three years of iterative improvement, the approach of FC has more advantages than TC’s in terms of course scores and student satisfaction. Computer programming courses are quite difficult and very rich in content. Class time is usually insufficient in the TC. However, the FC pedagogy can free a lot of class time. Therefore, the lecturer can cover more course material in the class and have more time for face-to-face discussion with students. In addition, the FC pedagogy can encourage students to become self-learners by pre-class active learning and help prepare them for how they will need to learn as computer engineers.

In K-12 context, the approach of FC is quite mature, with a large number of successful cases and relevant teaching data. However, in higher education, especially in computer programming courses, there are few approaches of FC for reference. When defining a new specific approach for FC in higher education, the following features of higher education should be considered.

1) College students are generally proficient in advanced technology. It is possible to use the latest technical tools to promote pre-class learning effect, such as online meeting, instant discussion group, online judge system, online questionnaire, etc.

2) College students are very busy and have a heavy workload. While encountering problems before class, they demand a faster response from teaching team. Establishing the course instant discussion group is an effective solution.

3) Higher education focuses more on ability training than just imparting knowledge. Lecturers could guide students to discuss more exploratory and expansive topics in class.

4) Through industrial project practice, it is helpful to cultivate students’ engineering ability. The progress inspection and problem solving in class can smooth the development of project.

5) Making full use of College Students’ fragmented time can help them save a lot of time. When preparing course materials, the learning duration of a single material should be controlled to 5-10 minutes. In addition, cellphone-based learning materials can strongly support active learning anytime, anywhere.

6) In college, teaching assistants are important members of the teaching team. The quantity and quality of teaching assistants are crucial to communicate and discuss with students instantly and effectively.

Therefore, the FC approach of this paper could be regarded as a reference for similar course in colleges. As mentioned above, the effect of FC approach depends too much on specific context to draw general conclusions.

VI. Conclusion

This paper reported the effects of the FC approach on course performance and student satisfaction across 2019-2021. Through a strict statistical analysis of teaching data in the three years, it was concluded that FC pedagogy could significantly improve course performance and student satisfaction. It was recommended to apply FC pedagogy in computer programming course of higher education. The approach of this paper could be used as an important reference. When defining a FC approach, students' education background and psychological feature should be considered and select the most appropriate ones for a particular context.

Future research will be carried out from three aspects. First, the effect of FC pedagogy will be continuously observed after the course material and teaching team is stable. Second, more studies will focus on the effect of students' engineering ability improvement. Finally, the course exam question bank will be developed. It can keep the exam difficulty roughly the same every year, so that the course performance in different years can be comparable.

Acknowledgment

This work was partially supported by the Beihang University education reform funding. We would like to thank the teaching assistants for their help in the implementation and improvement of the course. We would also like to thank the anonymous reviewers for their thoughtful feedbacks on this paper.

References

[1] J. Bergmann and A. Sams, Flip your classroom: reach every student in every class every day. Eugene, Or: International Society for Technology in Education, 2012.

[2] Bonwell, C. C., Eison, and J. A., Active Learning: Creating Excitement in the Classroom. Washington, D.C: School of Education, George Washington Univ, 1991.

[3] C. Papadopoulos and S. A. Roman, "Implementing and inverted classroom model in engineering statics: Intial results," in Proc. ASEE Annu. Conf., Louisville, KY, USA, 2010, p. AC2010-1868.

[4] B. Tucker, "The Flipped Classroom:Online instruction at home frees class time for learning," Education Next, 12(1), 82-83, 2012.

[5] C. K. Lo, C. W. Lie, and K. F. Hew, "Applying 'first principles of instruction' as a design theory of the flipped classroom: Findings from a collective study of four secondary school subjects," Comput. Educ., vol. 118, pp. 150–165, Mar. 2018.

[6] Ash, Katie, "Educators View the 'Flipped' model with a more critical eye", Education Week, 32(2), S6-S7, 2012.

[7] H. Al-Samarraie, A. Shamsuddin, and A. I. Alzahrani, "A flipped classroom model in higher education: a review of the evidence across disciplines," Education Tech Research Dev, vol. 68, no. 3, pp. 1017–1051, Jun. 2020, doi: 10.1007/s11423-019-09718-8.

[8] Betihavas, V., Bridgman, H., Kornhaber, R., and Cross, M. (2016). The evidence for 'flipping out': A system- atic review of the flipped classroom in nursing education. Nurse Education Today, 38, 15–21.

[9] Khanova, J., McLaughlin, J. E., Rhoney, D. H., Roth, M. T., & Harris, S. (2015). Student perceptions of a flipped pharmacotherapy course. American Journal of Pharmaceutical Education, 79(9), 140.

[10] Zainuddin, Z., Attaran, M. Malaysian students' perceptions of flipped classroom: A case study. Innovations in Education and Teaching International, 53(6), 660-670, 2016.

[11] Zappe, S., Leicht, R., Messner, J., Litzinger, T., and Lee, H. W. "Flipping" the classroom to explore active learning in a large undergraduate course. Paper presented at the American Society for Engineering Education, 2009.

[12] Velegol, S. B., Zappe, S. E., and Mahoney, E. The evolution of a flipped classroom: Evidence-based recommendations. Advances in Engineering Education, 4(3), 1–37, 2015.

[13] Amresh, A., Carberry, A. R., and Femiani, J. Evaluating the effectiveness of flipped classrooms for teaching cs1. Paper presented at the 2013 IEEE frontiers in education conference, 2013.

[14] O. Vojinovic, V. Simic, I. Milentijevic, and V. Ciric, "Tiered Assignments in Lab Programming Sessions: Exploring Objective Effects on Students' Motivation and Performance," IEEE Trans. Educ., vol. 63, no. 3, pp. 164–172, Aug. 2020, doi: 10.1109/TE.2019.2961647.

[15] R. Kay, T. MacDonald, and M. DiGiuseppe, "A comparison of lecture-based, active, and flipped classroom teaching approaches in higher education," J Comput High Educ, vol. 31, no. 3, pp. 449–471, Dec. 2019, doi: 10.1007/s12528-018-9197-x.

[16] A. E. Chis, A.N. Moldovan, L. Murphy, P. Pathak, and H. Muntean, "Investigating Flipped Classroom and Problem-based Learning in a Programming Module for Computing Conversion Course," Educational Technology and Society, vol. 21, no. 3, pp. 232–247, 2018.

[17] C.F. Lai, H.X. Zhong, and P.S. Chiu, "Investigating the impact of a flipped programming course using the DT-CDIO approach," Computers & Education, vol. 173, p. 104287, Nov. 2021.

[18] H. Y. Durak, "Modeling Different Variables in Learning Basic Concepts of Programming in Flipped Classrooms," Journal of Educational Computing Research, vol. 58, no. 1, pp. 160–199, Mar. 2020, doi:10.1177/0735633119827956.