Combined Problem-Based Learning in Computer Fundamentals Course

Changlong Gu and Xiaoying Li(✉)

College of Computer Science and Electronic Engineering, Hunan University, Changsha 410012, China
{guchanglong,lixy}@hnu.edu.cn

Abstract. The computer fundamentals course is an important course aiming at cultivating students' computational thinking. In order to improve the course teaching effects, combined PBL, which is an effective instructional approach that can help students to acquire knowledge and to master skills, was employed from 2021's spring semester to 2023's spring semester. The role of the teacher has changed from indoctrinator to facilitator. The learning model has shifted from being teacher-centered to a student-centered approach. Teachers carefully prepared various problems, applied new teaching pattern throughout the whole teaching process and gathered students' reflections. Through these reflections, this model's teaching effectiveness was verified. On the basis of problem solving, the ability of computational thinking and collaborative learning of students was improved. Furthermore, the conclusion summarizes the main ideas, and further work perspectives.

Keywords: Problem-based learning · Computer fundamental course · Computational thinking · Collaborative learning

1 Introduction

With the development of new technology trends, such as information technology, communication technology, operation technology, internet of things, and artificial intelligence, the age of Industry 4.0 has arrived. It is changing, at a deep level, people's way of working, living, communicating and learning. As a result, computer fundamentals courses are facing many challenges related to teaching content, teaching method, and teaching with technology. Moreover, traditional teaching content and teaching methods cannot meet the needs of the new era. In fact, the teaching content needs to change from software operations to computational thinking. It should heighten students' computational thinking abilities to thinking like computer scientists, improve computer literacy, and provide students with professional knowledge and professional ability [1]. The learning effectivity of traditional teacher-centered teaching methods is not efficient and cannot achieve the goal of raising students' computational thinking awareness.

Problem based learning (PBL) is an instructional design principle that describes a learning scenario where students are engaged in solving meaningful real-world problems.

W. Hong and G. Kanaparan (Eds.): ICCSE 2023, CCIS 2024, pp. 197–210, 2024.
https://doi.org/10.1007/978-981-97-0791-1_17

This process is generally divided into three stages: the problem analysis phase, learning knowledge and problem-solving phase, and reporting phase [2]. During the problem analysis phase, teachers interact with students, and together, they identify tasks to be accomplished, and lay down solutions that contribute to solving problems through acquiring new knowledge or applying existing knowledge. Regarding the stage of learning knowledge and problem solving, students learn with problems to be solved in mind. Therefore, students acquire knowledge on an individual basis or from peers through small-group discussions. In the reporting phase, students consolidate their learning through reflective writing.

PBL is consistent with constructivist learning theory. Problems provide a setting for students to build constructive knowledge and acquire problem-solving abilities. Also, PBL allow students to develop critical thinking and communication skills. Therefore, under PBL, students can actively participate, experience deeply, and establish their own learning methods for knowledge acquisition and understanding throughout the process. PBL is motivational, supports students' socialization, thinking, and self-regulation skills [3].

The flipped classroom was first used by Jonathan Bergmann and Aaron Sams, who were high school chemistry teachers from Colorado in 2006 [4]. In the flipped classroom method, the lecture happens out of class, while the practical application assignments, formerly known as homework, take place in the classroom [4]. The studies on the flipped classroom indicate that flipped classroom has a positive impact on students' learning and retention [5] and can improve students' critical thinking skills [6].

Collaborative learning is a form of group learning in which groups are formed around a central topic. As a result, group members exchange their experiences and opinions, discuss problems, request from each other a piece of advice, discuss with each other, help each other, construct knowledge together, and jointly accomplish learning goals. Collaborative learning is regarded as "suggesting a way of dealing with people which respects and highlights individual group members' abilities and contributions" [7].

In this paper, the authors present the new course content and a combined PBL teaching method which combining the PBL approach, the flipped classroom and the group collaborative learning. Additionally, the authors provide the collected students' reflections, summarize the results, and discuss the major issues to focus on when implementing such a model. As main contribution, this work depicts the effect of a combined PBL strategy in a university computer fundamentals course over students' learning. Moreover, it investigates a specific pedagogical approach aiming at assessing the validity and improving the quality of teaching and learning of the computer fundamentals course.

2 Status Analysis

2.1 Problems in the Teaching Content

Most existing teaching contents related to non-major computer fundamental course include several topics, namely, the introduction to computer composition and principle, the introduction to several commonly used software applications, or the learning of programming languages such as C, C + +, VB, or Java for the different non-majors. With the advent of the AI era, relying only on such content is no longer enough. Learners are eager to learn AI technology. Moreover, learners should be taught on how to use the computer as a tool, as well as to solve problems and to think like a computer scientist. For most non-major computer students who learn C/C + + /Java languages, due to the disconnect between what they have learned and the application of such knowledge into their respective majors, they can straightforwardly feel that they have "learned something useless". The programming language in the teaching content should be conducive to learn AI technology. However, the Python language is such a language, which is simple and easy to use, and can helps users focus on the solution to their real-world problem and not the structure or syntax of the problem [8]. It supports many libraries and has good computing ecology.

2.2 Problems in the Teaching Process

In traditional learning, teachers always impart knowledge in class and assign jobs with specific requirements or expected outcomes after class. Teachers always teach the same information to all students, even if their foundations and abilities are different [9]. In such a traditional paradigm, teachers are in charge of learning, while students passively receive knowledge from their teachers. Information is passed only from teachers to students, with little or no feedback from students to teachers. There is little to no communication or discussion among students. However, under such a course, over time, students can easily become tired or bored.

Students' problem solving and creative thinking skills are generally insufficient to meet job requirements. Therefore, when encountering a real-world problem, students are unable to arrive at a solution. Students often encounter various difficulties when completing a slightly more complex homework assignment. Many of them need to spend a lot of time completing their assignments, and some students may not finish their assignments on time. As the difficulty of the course increases, those students gradually lose their interest in the course.

Under the traditional teaching model, students do not have sufficient opportunities to develop their critical thinking and problem-solving skills due to limited learning time and the use of a teacher-centered model. The goal of a computer fundamentals course does not revolve only around learning knowledge, but more importantly around exercising the programming ability. Moreover, because the concept of programming requires many skills and a broad knowledge, students always give up at some point of the programming learning curve and the goals of the course are always missed.

3 Combined PBL in Computer Fundamental Course

A PBL approach combined with the flipped teaching method and collaborate learning method has been developed to improve the learning efficiency of computer fundamentals courses as well as to solve the drawbacks in the traditional teaching and learning model [10, 11]. Under such a combined approach, questions or tasks assigned by teachers are carried out in class and completed under the guidance of teachers. Additionally, some programming training is completed in class and students are requested to analyze these questions and to spare no effort in finding the solutions. As a result, students can identify what knowledge is necessary, and can acquire such knowledge by themselves or through cooperation with each other. Students learn by doing and study while practicing until they can solve these problems. Moreover, students need to communicate with teachers and peers in real time, which has as an outcome high students' interest in the course. Figure 1 depicts the differences between traditional learning and combined PBL.

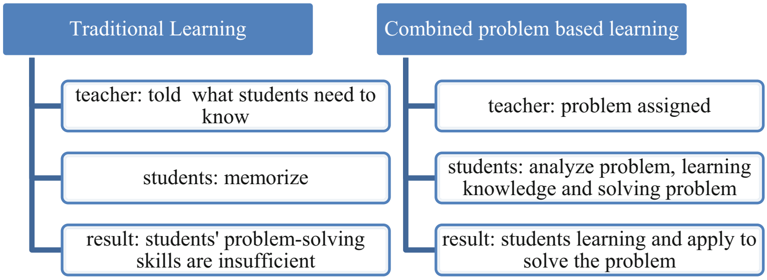

Fig. 1. Difference between traditional learning and combined PBL

In the combined approach the role of the teacher has changed from indoctrinator to facilitator [12], and the teaching mode has changed from "indoctrination" teaching to designing a motivating problem or question for students to investigate, as well as to guide them on how to learn. Throughout the entire process, teachers need to guide students' discussion, encourage students to think creatively, keep their interest at a high peak, provide analysis or pieces of advice, and evaluate students' learning abilities. Moreover, teachers always need to design specific learning resources or to provide existing ones. Additionally, teachers need to summarize the process of teaching and learning and assist in the process of building up students' collective knowledge with open-ended metacognitive questions [13]. An effective PBL-CT integration enhances teaching and learning besides improving students' computational thinking skills [14].

4 Introduction to Computing and Artificial Intelligence Course Descriptions

4.1 Course Setting

The computer fundamental course plays an important role in cultivating students' information literacy in universities. Such an affirmation makes more sense when considering the development of big data, cloud computing, internet of things, AI and other new technologies. As a result, the teaching contents and teaching modes of computer fundamental courses in universities are changing.

The course denoted as Introduction to Computing and Artificial Intelligence is a public basic course of information technology in Hunan University. This course is designed for freshmen students from non-computer majors to cultivate their computational thinking—since the 2020 fall semester. The teaching contents of this course include the Python language foundation, computational thinking and artificial intelligence algorithms and their applications. It can be divided into seven modules, as shown in Table 1. The Python language is used as the main programming tool to realize different thinking modes of computational thinking, which include computing system thinking, network thinking, data thinking, algorithm thinking and artificial intelligence thinking. These are viewed as the core components of computational thinking. Therefore, from the perspective of computational thinking, such a course is guided by artificial-intelligence-related problems and is taught through problem solving. As a result, students are being trained to master the solution consisting of traversing from a professional problem under a computing system to an algorithm first, then second to a program.

Table 1. Content of Introduction to computing and artificial

Module name	Description	Theory time	Experiment time
Overview of computing and computational thinking	History of computing, concepts of computational thinking	2 h	
Programming fundamentals	Python programming foundation	16 h	16 h
Computing system thinking	Mathematical Logic, Components of Computing Systems	2–4 h	2–4 h
Networking thinking	Internet, network spider, data parsing	2–4 h	2–4 h
Data thinking	Data management, data cleansing, data analysis and data visualization	4–10 h	4–8 h
Algorithm thinking	Classical algorithms	2–10 h	2–6 h
AI thinking	Machine learning, deep learning, large model applications	8 h	4 h

Different majors focused on different modules in their teaching, and the learning hours of each module were different.

4.2 Teaching Approach

In the 2020 fall semester, the traditional teacher-centered approach was used in the course. However, cultivation of computational thinking can be a difficult undertaking for some. For example, abstraction skills, decomposition skills, pattern recognition skills, programming skills, and logical reasoning skills are needed [15]. The teaching result was not satisfactory.

In the 2021 spring semester, a combined PBL approach was employed in the course. An online and offline blended teaching mode was adopted. Some parts of the concepts of computational thinking and artificial intelligence can be learned online, while the other parts are learned offline. Such a course emphasizes the cultivation of computational thinking ability, and the teaching method employed has also changed from "knowledge output" to "ability training". As a result, students are trained through PBL with intensive teaching and practice and learning by doing. Moreover, students conduct many standalone short problems and undergo problem-targeted training under team collaboration through teaching platform. They can practice through games in experimental class settings under a teacher's guidance. Therefore, through team training, students' cooperation consciousness and innovation ability are improved. Additionally, students' problem-solving ability using computational thinking is developed at multiple levels. Such a course will lay a foundation for students of various majors to design, construct and apply various computing systems to solve disciplinary problems in the future, as well as help learners improve their ability to interpret real-world systems and solve complex problems.

Multi-dimensional learning assessments is used to evaluate student's final score. This evaluation method reflects the learning process employed, such as short problems scores, student–student mutual evaluation scores, unit test scores, team comprehensive assignment scores, mid-term exam and final exam, as shown in Table 2. Considering Table 2, usual performance 1 usually groups together short problem scores in classroom performance and team comprehensive assignment score, while usual performance 2 consists of experiment scores, quiz scores and unit test scores.

Table 2. Evaluation method of computer fundamental course

Evaluation indicators		Percent of the total score	
Usual performance 1	short problem practices	10%	20%
	team comprehensive assignment	10%	
Usual performance 2	experiment	15%	30%
	quiz	5%	
	unit test	10%	
Midterm examination		10%	
Final examination		40%	

5 Combined PBL Implementation Descriptions

5.1 Planning

Non-computer-major students have enrolled in the Introduction to computing and artificial intelligence course since 2020 fall semester. Over 80% of them had no prior coding experience, according to a questionnaire survey of 721 freshmen in the fall semester of

2020. From the 2021 spring semester, the combined PBL method was adopted. In the 2021 spring semester, about 86 students enrolled in the Introduction to Computing and Artificial intelligence course participated willingly in the experiment.

Due to the zero-coding level of most students, we carefully prepared various problems for the course from a simple level to a complex one. The problems related to instructional design followed the same principle. These problems were classified into two types. One type was short problems, the second type was about large real-world problems. For each lecture class, we prepared about four short problems. Totally 60 short problems were prepared. Additionally, we prepared 4 large comprehensive problems from the real world for students to choose from. In the 2022 fall semester, the number of large problems added to 10, as shown in Table 3. Short problems were suitable for a single person to work on, while large problems were complex problems more suited for group collaborative learning activities.

Table 3. Evaluation method of computer fundamental course

Large problems	Part of Short problems
E-commerce marketing practice project - Construction and Evaluation of a model for predicting mobile phone sales	Currency conversion
Calculation and visualization of drug molecular fingerprints	Wheat on the chess board
Analysis of listed company stocks	Extracting information from ID card number
Analysis and management of large ancient poetry datasets	Caesar code encryption
Milk tea shop service system	Hurricane classification determination
Book management system	Mars rover statistics
Recruitment website information crawling and analysis	Random password generation
High quality film data acquisition and analysis	Word cloud production
Content crawling and analysis of literary works	Shooting curves of basketball drawing
Practical project for the development of Gobang game	Face detection

The course team teachers produced 66 micro lesson videos for students' preview and review. Those videos were published on the Educoder platform (https://www.educoder.net). Educoder is a network teaching and learning platform that provides many functions, such as course creation, experiment creation, automatic score judgement, real-time evaluation, learning resource presence, and data statistics. Students can practice repeatedly on the platform. Additionally, the platform supports exercise banks and examination question banks, as well as examination functions, and digital teaching management. Teachers build and publish various course resources through the platform and track students' learning activities throughout the process. We create an online teaching course on the Educoder network platform, providing experiments and other learning resources. As a result, students can watch these videos from anywhere and at any time. Corresponding quizzes were also prepared with these videos. Courseware, exercises and unit tests were also prepared. We also created 15 learning guides, 20 courseware items, 16 experiments, 9 after-class exercises, and 6 tests. All of these teaching and learning resources were published on our online course in Educoder.

5.2 Implementation

At the beginning of the course, students were divided into groups. Each group consisted of three to four students. Students were required to preview the course content before the lecture class on the Educoder network platform according to the teacher's guidance plan. These learning resources include lecture videos, courseware, etc. As a result, students acquired basic knowledge or concepts before the lecture class.

The lecture class was divided into three parts, with each part having specific objective, namely, the preview test, problem-based learning, and the summary. A multiple-choice quiz was given onto the Educoder network platform at first, which would take about 5min. After the quiz, the Educoder automatically judged the answers and provided score statistics. The teacher could immediately check the score of each student and the statistical data of all the students on Educoder, including the highest score, the lowest score, the average score, the accuracy rate of each question, etc. The teacher then analyzed the different questions with students. Such process required around 5–10 min.

Then, the teacher introduced about short problems. Students discussed their understanding regarding a specific problem using flowcharts and discussed ideas and methods of solving problems in groups. They acquired the necessary knowledge by studying in a collaborative manner. When a group met difficulties, the teacher provided clues. However, students were required to program independently and complete their own summary reports. At the end of the class, students submitted their work on the Educoder, then the teacher summarized and displayed excellent work. After the class, students completed assignments and extended their knowledge and skills. Each student had to submit their learning report, which included the problem solution, program code and learning harvest. Following report submission, students gave a peer assessment to other students. From the peer assessment, students could learn different solutions to the same problems. The teaching process is shown in Fig. 2.

While in practice classes, students completed experiments on the Educoder platform. Unit tests were also conducted in practice classes to check students' performance. The measured students' performance could affect the future choice of the teacher's strategy.

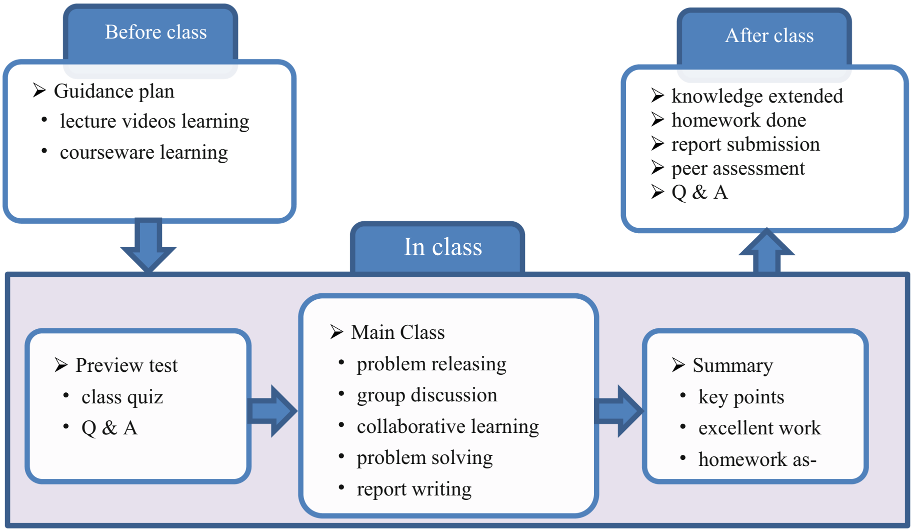

Fig. 2. Implementation of combined PBL approach

5.3 Team Comprehensive Training Problems

The large problems were assigned at the beginning of the semester. The large problems integrated multi-methods of computational thinking and required students to complete them in groups. Each group chose one large problem to design and implement according to their interest. Students could also choose another artificial intelligence problem according to their interest and ability.

When groups had determined the problem, the next step was to clarify the division of each group member. For example, one person in the group was responsible for scraping data from the website and saving the obtained data as a file, while the second person was responsible for data cleaning, sorting and analysis. The third group member was responsible for data visualization. Group members discussed what data to scrap from what website, as well as what kinds of analysis and visualization are suitable for the obtained data. Furthermore, as requirements, at least two analysis and visualization results were required to be displayed. Finally, other members used artificial intelligence model for modeling and prediction. Each member of the group then implemented his/her module with a Python program and combined it into a complete program. Finally, each group wrote a report that contained the description of all the work completed.

The large problem needed to be solved out of class. This required close cooperation among group members. If a student encountered difficulties, he/she could discuss and learn how to solve them in his/her group or seek help from the teacher. After all group reports had been submitted on Educoder, teachers gave scores to each group. The final score of the large problem was made of the scores related to coding, functions, and reports. Then, the teacher chose excellent works to review and share. This activity will further promote students' learning interests and abilities.

6 Students' Reflections and Analysis of the Teaching Effect

To investigate the effectiveness of the combined PBL approach adopted in the course in the spring semester of 2021, students' reflections from this course were gathered. Those reflections include the study status and effect, as well as group collaborative effect, which come from two questionnaires and students' large question report harvests. In week 7, a survey was conducted through a questionnaire. Seventy-six students participated in the questionnaire survey. In such questionnaire, there was an open-ended question "Please comment on your study this week." Nine students were not satisfied with their status and thought they should devote more time to the course. Fourteen students thought their learning status was average. Due to the complexity of the learning content, they should practice more. The remaining fifty-three students (70%) thought they had made progress. Part of the answers were as follows.

"It is better than last week. We have more communication with group members and give better play to the role of group cooperation."

"The learning status and effect are good. We can actively analyze problems and communicate with team members, which is reflected in more active questioning and communication with team members and master more list related knowledge."

"In this week's study, through many exchanges with teacher, teaching assistants and students in team, including the study of relevant videos before class, I found many deficiencies in my study."

Concerning the effect of teamwork towards the PBL approach in this course, about 82% of students thought it worked out, and were willing to communicate and help the students in the group. Figure 3 displays the response to the problem "Do you think teamwork has worked this week?".

Pre-class previews require students to learn specific knowledge before the class, which can reflect students' learning status and participation in learning. In the teaching survey of week 11, 91% of students carried out the pre-class preview. The group learning effectivity of the large problem was expressed from the students' group report harvests. The features of PBL such as creativity, collaboration and critical thinking impressed students deeply. Some reports written by students were as follows.

"This time, I have two harvests: one is the power of the team; the other is the power of innovation and creation - these two powers support us from facing the assignment requirements, to generating initial ideas, to the idea of from the skeleton to the full."

"In the teamwork, I have also gained and grown. What impressed me especially is the strong execution of my two teammates. Everyone has a clear division of labor, each performed their own duties and helped each other, which was the key to the success of our team."

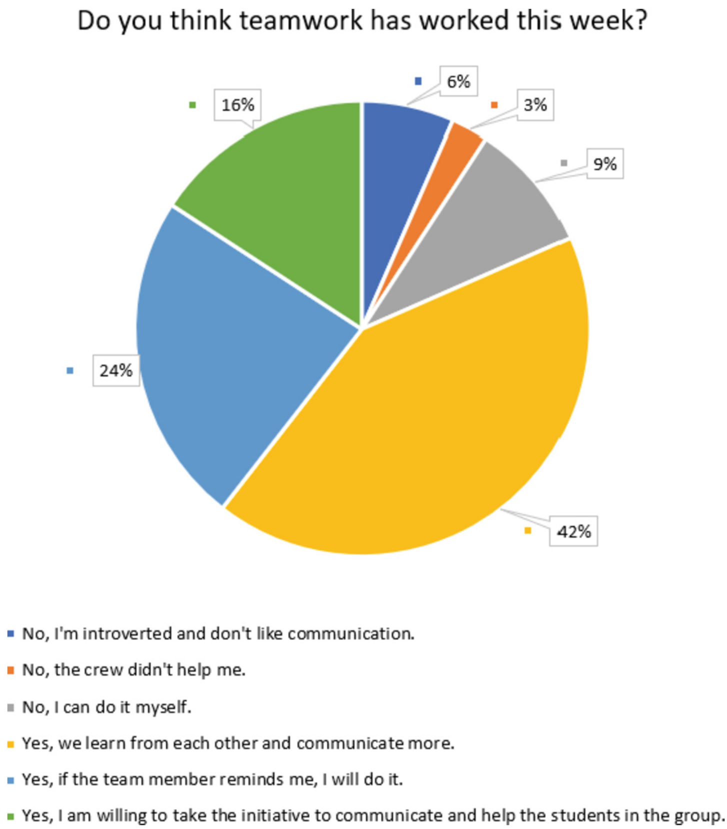

Fig. 3. Pie chart of question about the effect of teamwork

To further investigate the effectiveness of the combined PBL approach, one class taught by the same teacher in the 2020 fall semester was compared with a class in the 2021 spring semester and a class in the 2023 spring semester. The number of people in the three classes is 70, 86, and 62, respectively. The final scores of the three classes are shown in Fig. 4. The 2020 fall semester did not adopt the combined PBL teaching method. These examinations were given according to the same teaching contents and for the same difficulty level. The passing rate of 2021 spring and 2023 spring semester's final scores were 93.3% and 100.0%, which were 0.7% and 7.4% higher than those of 2020 fall. Also, the excellence rate (90–100) of 2021 spring semester's and 2023 spring semester's final examination were 8.3% and 29%, which were 5.4% and 26.1% higher than 2020 fall semester's final scores. From the comparison of the average grades of the three semesters (see Fig. 5.), it can be seen that the grades are getting better and better, and to some extent, the teaching effect is getting better and better.

According to the curriculum teaching evaluation conducted by the university at the end of the two semesters, the students' satisfaction rate with the curriculum teaching effect has increased from 79.41% to 89%.

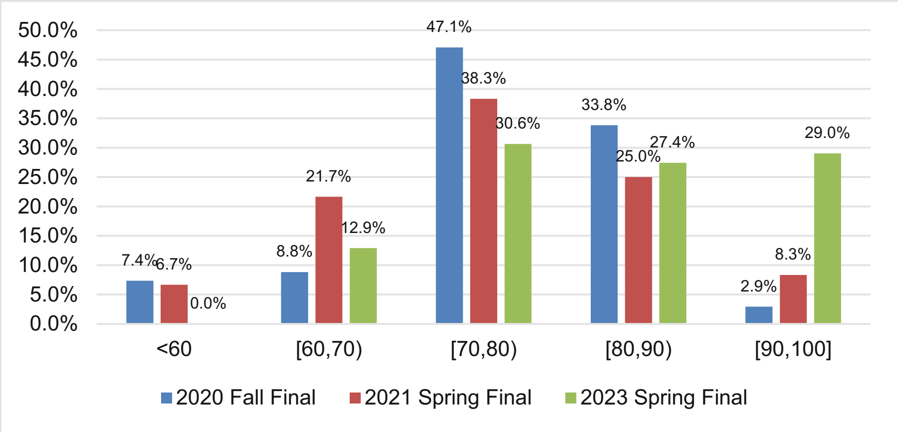

Fig. 4. Comparison of Fall 2020, Spring 2021 and Spring 2023 Final

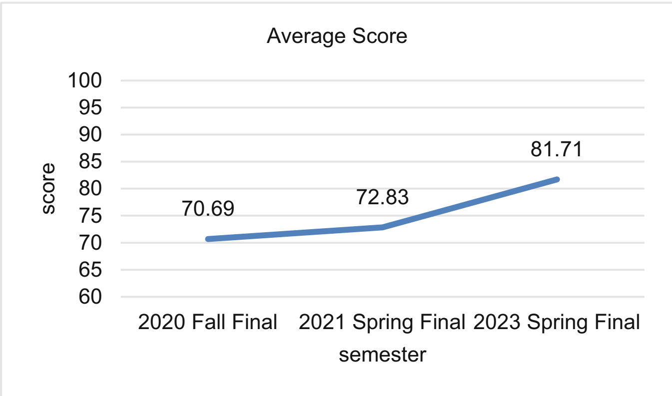

Fig. 5. Comparison of class average score in Fall 2020, Spring 2021, and Spring 2023

7 Conclusions

In the Introduction to Computing and Artificial Intelligence course, the teaching goal became to cultivate students' computational thinking ability based on problem solving. Through the teaching experience of using the combined PBL approach since 2021 spring semester, the effectiveness of combined PBL was proved. It enhanced students' initiative and enthusiasm for learning. Students' learning statuses were always well preserved. As a result, group learning promoted learning effectiveness, collaboration, critical thinking, and creativity.

Despite the potential benefits of incorporating a combined PBL approach in our computer fundamentals course, there are still limitations that may hinder its effectiveness and quality. One such limitation is the need for carefully crafted problems that progress from simple to complex. It is advisable to avoid incorporating too many specific grammatical details in problem design, as this knowledge can easily be acquired online and quickly resolved through student collaboration. Additionally, teachers must ensure that class progress remains on track to avoid delays in completing subsequent teaching tasks.

Although the combined PBL has achieved good results, 82% of respondents (Fig. 3) benefited from teamwork. However, there is still 18% who did not benefit from the application of the combined PBL method. In China, the Outline of the National Medium- and Long-Term Education Reform and Development Plan (2010–2020) points out that it is necessary to care of each student, respect the law of physical and mental development of students, and provide suitable education for each student [16](Han & Ye, 2017). There are differences in learning ability, hobbies, ways of thinking, cognitive characteristics, etc. Some methods should be adopted to help these students. In the future, personalized learning according to students' individual differences and learning needs should be carried out. It can help students to find learning styles adapted to them.

References

1. Wing, J.M.: Computational thinking. Commun. ACM **49**(3), 33–35 (2006)
2. Schmidt, H.G., Moust, J.H.: Factors affecting small-group tutorial learning: a review of research. Probl.-Based Learn.: Res. Perspect. Learn. Interact., 19–52 (2000)
3. Dilekli, Y.: Project-based learning. In: Paradigm shifts in 21st Century Teaching and Learning, pp. 53–68. IGI Global (2020)
4. Arnold-Garza, S.: The flipped classroom teaching model and its use for information literacy instruction. Commun. Inf. Lit. **8**(1), 9 (2014)
5. Mithun, S., Evans, N.: Impact of the flipped classroom on students' learning and retention in teaching programming. In: 2018 ASEE Annual Conference & Exposition (2018)
6. Davenport, C.E.: Evolution in student perceptions of a flipped classroom in a computer programming course. J. Coll. Sci. Teach. **47**(4), 30–35 (2018)
7. Laal, M., Laal, M.: Collaborative learning: what is it? Procedia Soc. Behav. Sci. **31**, 491–495 (2012)
8. Python Geeks. What is Python Programming Language? https://pythongeeks.org/what-is-python-programming-language
9. Berssanette, J.H., de Francisco, A.C.: Active learning in the context of the teaching/learning of computer programming: a systematic review. J. Inf. Technol. Educ. Res. **20**, 201 (2021)
10. Wang, G., Zhao, H., Guo, Y., Li, M.: Integration of flipped classroom and problem based learning model and its implementation in university programming course. In: 2019 14th International Conference on Computer Science & Education (ICCSE), pp. 606–610). IEEE (2019)
11. Chis, A.E., Moldovan, A.N., Murphy, L., Pathak, P., Muntean, C.H.: Investigating flipped classroom and problem-based learning in a programming module for computing conversion course. J. Educ. Technol. Soc. **21**(4), 232–247 (2018)
12. Frydenberg, M., Mentzer, K.: From engagement to empowerment: project-based learning in Python coding courses. In: EDISG Conference, Information Systems & Computing Academic Professionals (2020)
13. Yew, E.H., Goh, K.: Problem-based learning: an overview of its process and impact on learning. Health Prof. Educ. **2**(2), 75–79 (2016)
14. Saad, A., Zainudin, S.: A review of project-based learning (PBL) and computational thinking (CT) in teaching and learning. Learn. Motiv. **78**, 101802 (2022)

15. Yusoff, K.M., Ashaari, N.S., Wook, T.S.M.T., Ali, N.M.: Analysis on the requirements of computational thinking skills to overcome the difficulties in learning programming. Int. J. Adv. Comput. Sci. Appl. **11**(3) (2020)
16. Han, S., Ye, F.: China's education policy-making: a policy network perspective. J. Educ. Policy **32**(4), 389–413 (2017)