IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol.005, LIIR193
DOI: https://doi.org/10.52731/liir.v005.193

An Attempt at Peer Learning with Explanation in Introductory Programming Education

Satoshi Cho*, Hisashi Sato *

Abstract

This paper introduces "Peer Learning with Explanation" (PLE), an innovative approach in introductory programming education, inspired by pair programming. PLE is a collaborative learning strategy where students work in pairs to solve programming tasks, with one student explaining their reasoning and methods to their partner. This approach aims to deepen understanding and enhance learning outcomes through the sharing of ideas and feedback. A key feature of PLE is its structured method for addressing exercise problems, clearly outlining the steps involved, what actions to take, what to explain, and what to discuss. This structured approach ensures that students are focused on their learning goals, thereby improving their comprehension and problem-solving abilities. The study, conducted with first-year students at the Kanagawa Institute of Technology, Department of Information Media, investigated the effectiveness of PLE compared to traditional solo programming methods. Results indicated that students engaged in PLE demonstrated a better grasp of programming concepts and skills. PLE encourages students to articulate their thought processes and problem-solving strategies, leading to a more profound understanding and retention of programming knowledge. It is also expected to foster improved communication and collaboration skills, which are essential in the programming field. The findings suggest that PLE can be a valuable tool in programming education, significantly contributing to students' learning experiences and outcomes.

Keywords: Education, Pair programming, Peer learning, Programming

1 Introduction

In recent years, amidst the rapid evolution of information technology and the digitalization of society, the importance of programming education has significantly increased. Many countries have integrated programming education into their foundational curricula, striving to develop programming skills among students [1][2]. Consequently, various effective methods and approaches for programming education are being explored.

In programming learning, conventional methods include lectures by teachers, solo programming for exercise tasks, project-based learning in groups or teams, and practical learning experiences such as hackathons. Among these, pair programming has been incorporated and has become a common learning method. It is often typical for beginners, who are just starting with programming and may vary in individual skill and understanding, to first acquire basic abilities through lectures and solo programming exercises before engaging in group work. Recently, the advent of generative AI, exemplified by OpenAI's ChatGPT, has demonstrated capabilities to the extent that it can produce solutions to beginner-level exercise problems simply from the input of

* Kanagawa Institute of Technology, Kanagawa, Japan

the problem statement. Originally, exercise problems are meant to be educational materials, and if someone else, including AI, substitutes the learning process, the intended educational benefits are not achieved. Some learners may focus more on clearing tasks rather than on genuine learning, leading to doubts among educators about the educational effectiveness of solo programming exercises. Therefore, to enhance learning efficacy, it is proposed not only to create and test exercise programs but also to include explanations of the programs.

This paper proposes a novel learning approach, "Peer Learning with Explanation" (PLE), inspired by pair programming, a well-known agile methodology. The effectiveness of pair programming in higher education for programming learning has been reported, indicating its benefits for a wide range of individuals, including high school and college beginners, students with some experience in programming, and working adults, irrespective of their age, position, or environment [3] [4]. Additionally, the promotion of knowledge sharing among students has been suggested [5].

The "Peer Learning with Explanation" (PLE) approach proposed in this paper is designed to address such challenges. PLE involves students working in pairs to collaboratively solve tasks, with the expectation of promoting deeper understanding through the sharing of ideas and feedback. Especially in the context of programming education, adopting this approach is anticipated to enhance the learning outcomes of students.

This study reconstructs pair programming for educational purposes in programming and proposes PLE, where students tackle exercise problems while explaining them, aiming to confirm its effectiveness in enhancing educational outcomes.

1.1 Related Work

In the field of programming education, recent years have seen a growing interest in studies related to pair programming and agile development methodologies. Moreover, peer learning, which involves a dyadic educational system, is closely related to this study. Below are some of the key research findings in these areas.

Peer Learning: Learning in pairs, a form often seen in collaborative learning across various fields beyond just programming, involves students taking on roles such as task executor and monitor, which deepens their thinking [6]. It has also been pointed out that pairs demanding explanations from each other and providing explanations are beneficial for collaboratively solving problems [7]. Incorporating pair programming as a form of peer learning and implementing flip teaching have also been reported. In such cases, students were asked to watch programming lecture videos before class and engage in pair programming during class. Analyses of Learning Management System (LMS) access logs, learning records, and surveys suggest that collaboration among learners with similar programming skills promotes deeper understanding of the content [8].

Research on Education and Pair Programming: Baichang Zhong and colleagues have investigated the impact of the duration of role-switching in pair programming. Their findings suggest that semi-autonomous switching is more effective for learning outcomes than fixed intervals [9]. Meanwhile, Onni Aarne and colleagues have explored the practice and effects of pair programming within the context of universities, where high student autonomy and personal responsibility are emphasized [3]. Their research reveals how gender and previous programming experience correlate with participation in pair programming labs. Deepak Kumar and others have conducted detailed investigations into how students practice agile development, including pair programming, within a learning environment. In particular, their analysis of challenges and experiences for both students and lecture staff revealed that some students, lacking in programming knowledge, required more specialized expertise in programming [10]. Kyungsub Stephen Choi has specifically researched the impact of gender in pair programming [11]. His study provides insights into how pairs of different genders collaborate and the resulting effects on learning outcomes and the quality of communication.

Through these prior studies, a deeper understanding is gained of peer learning, pair programming, and agile development methodologies in education, particularly regarding the impact of student backgrounds and gender on learning outcomes. This research builds upon these studies, attempting an analysis from a new perspective.

2 Proposed Method

Pair programming, a known software development technique, involves two developers collaboratively writing code on a single computer. Here, one acts as the "Driver," directly writing the code, while the other serves as the "Navigator," monitoring and providing feedback. This approach enhances code quality, early error detection, and knowledge sharing [12]. The Driver handles tasks such as coding and testing, while the Navigator oversees the Driver's code, offering feedback and advice, and guiding the design and direction, thinking ahead about next steps and potential issues.

PLE adapts this collaborative learning form and the pair programming method for educational settings. As highlighted in related studies, attempts have been made to utilize pair programming in education. Unlike its software development counterpart, focusing on software quality and efficiency, PLE aims at deepening students' understanding and improving learning outcomes. In PLE, students pair up, with one taking the role of a "Driver" and the other as a "Navigator," mirroring the pair programming roles but adjusted for an educational context. The Driver, while tackling exercise problems, explains their thought process and programming approach to the Navigator, jointly exploring solutions. A significant difference between PLE and pair programming is that the Driver also considers the design and direction of the problem, communicating and validating these with the Navigator. (Figure 1)
PLE is envisioned for lecture and practical class formats, with the expectation that students engaging in PLE during practical sessions will be more proactive in attending lectures compared to solo programming.

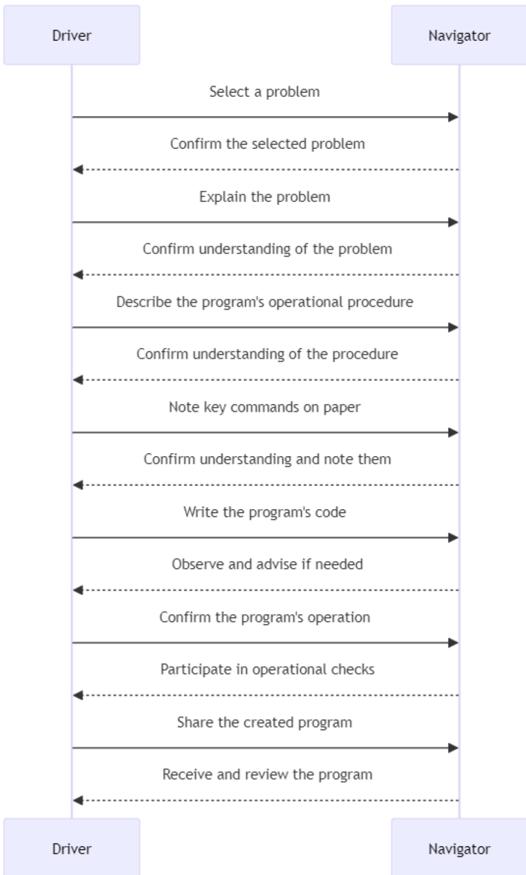


Figure 1: Driver and Navigator sequence

2.1 Formation of Pairs and Role Rotation

In implementing PLE, students form pairs through various methods, often designated by the instructor. However, random pairing or pairing based on students' abilities and experiences are also considered. Regular changing of pairings is recommended, enabling students to gain diverse collaborative experiences. Studies indicate that differences in gender [3] or personality [4] do not affect learning efficiency, suggesting that random pairings are sufficiently effective.

During PLE, students switch between the Driver and Navigator roles after each exercise problem. This rotation aims to ensure that students equally experience both roles, maximizing learning from each perspective. This role rotation is also expected to enhance the quality of student communication and cooperation, with reports suggesting increased student satisfaction and reduced frustration [13].

2.2 Role of the Driver

The Driver interprets programming problems, managing each process from design to implementation. Before proceeding, the Driver explains the process and its intent to the Navigator, who understands and clarifies any uncertainties, then approves the progression. This mutual confirmation and approval deepen understanding and aim for higher-quality program development.

Instructions given to the Driver include "using the PC," "explaining to the Navigator upon inquiry," "collaborating with the Navigator in documenting the process," and "seeking advice from the Navigator when stuck."

2.3 Role of the Navigator

The Navigator questions the Driver when unclear about explanations and documents the Driver's thought process, making it visible and shared. When consulted by the Driver, the Navigator collaborates to find solutions.

Navigator instructions include "not using the PC," "immediately questioning any uncertainties in the Driver’s explanations," "briefly documenting the Driver’s explanations on paper," and "thinking together with the Driver when consulted."

2.4 Instructor's Role

While primarily observing, instructors intervene with advice when pairs face difficulties in problem-solving. Solo programmers might neglect challenging problems or seek solutions far removed from the learning objective. Pairs prevent such situations and encourage students to consult instructors without hesitation.

2.5 Aims of PLE

The primary goal of PLE is to share the problem-solving process among students, deepening mutual understanding. By explaining each process's content and intent, the Driver clarifies their own thinking, and the Navigator, by understanding and approving, helps both achieve a deeper comprehension of problem-solving methods and program operation. This mutual confirmation and approval prevent superficial understanding, like "the program just works somehow." PLE is also expected to improve communication and cooperation among students. By necessitating explanations for exercise problems, students are anticipated to attend lectures more effectively, with a clearer purpose.

2.6 Steps of PLE

A key feature of PLE is the Driver explaining the approach to the Navigator, who must understand and agree. For beginners in programming, tackling exercises without clear explanations or prematurely starting coding can misalign efforts. This study, therefore, outlines steps for tackling exercises, where the Driver explains each step to the Navigator, who then confirms understanding and agreement. These steps (Table **1**) differentiate PLE from pair programming and are central to our method.

Each exercise problem follows seven steps, with roles swapping after completion.

Table 1: Steps of Peer Learning with Explanation

Step	Driver	Navigator
1	Select a problem	Confirm the selected problem
2	Explain the problem	Confirm understanding of the problem
3	Describe the program's operational procedure	Confirm understanding of the procedure
4	Note key commands on paper	Confirm understanding and note them
5	Write the program's code	Observe and advise if the Driver hesitates
6	Confirm the program's operation	Participate in operational checks
7	Share the created program with the Navigator	Receive and review the program

This table closely resembles the one provided to learners. Steps 2-6 particularly define the PLE approach.

Unlike simply inputting and modifying sample codes provided in lectures, PLE's initial steps (2, 3) consolidate a conceptual understanding of the content. The later steps (4, 5) involve crafting specific operations and codes, thus ensuring a more structured, comprehensive learning approach.

3 Research Methodology

The primary objective of this study is to evaluate the effectiveness of Peer Learning with Explanation (PLE) in programming education. Specifically, we adapted the agile methodology of pair programming to the educational context, examining the educational impact of explaining programming exercise problems to peers. The study aims to uncover whether this method deepens beginners' understanding and enhances their learning outcomes. Additionally, we provide a detailed analysis of the characteristics and effects of PLE by comparing it to traditional solo programming practices.

3.1 Background of the Experiment

This study was conducted as part of the "Introductory Lecture of Information Systems and Media Design," a course taken by first-year students of the Department of Information Media at the Kanagawa Institute of Technology's Faculty of Information Technology. This foundational course, the first of its kind for students in the Information Media program, aims to cultivate core technologies for creating and disseminating information through programming.

The structure of the classes involves a 90-minute lecture, followed by a 180-minute practical session held after a two-day break. In these sessions, students engage with exercise problems that are made available on the same day. Instructors oversee the proceedings while also addressing inquiries from the students. During these practical sessions, every student is presented with identical problems through the LMS (Learning Management System), and their responses are submitted online. This means that the exercise problems are uniform for all students. In the final class session, a comprehensive exam is conducted, where all students are posed the same questions.

Information about the PLE methodology is available on a web page, allowing students to access and review the content at any time.

3.2 Programming Language Used

In the experiment, the programming language "Processing" was employed. Based on Java syntax, Processing is particularly well-suited for visual expressions and the development of interactive applications. Its simplicity and intuitive nature make it an accessible language for beginners, and it is widely used in educational settings.

During the lessons, Processing is taught with an emphasis on structured programming rather than object-oriented programming. Students use this language to learn foundational operations like drawing shapes, displaying images, and creating interactions using keyboard and mouse inputs. Starting with such basics enables students to grasp fundamental programming concepts and thought processes.

3.3 Experimental Setup

The study was carried out with first-year students from the Department of Information Media at the Kanagawa Institute of Technology's Faculty of Information Technology, spanning from April to July 2023. A total of 202 students, including new enrollees and a few re-enrollees, participated. They were divided into five classes (A, B, C, D, E), each adopting different learning methodologies. Specifically, only students in Class A utilized the PLE method, while those in Classes B to E engaged in solo learning for their exercise tasks.

In all classes, students were allowed to consult with each other and ask questions of the instructors, who could also add explanations to the exercise problems as they deemed necessary.

Pairings in PLE were randomly assigned by instructors, regardless of the students' abilities or experience. This random pairing was intended to ensure a fair evaluation of the PLE's effectiveness.

For data collection, final exam grades were used as the primary metric to assess the impact of PLE. The performance of students in Class A, where PLE was implemented, was compared with that of students in Classes B to E, where it was not.

3.4 Data Collection

In this study, the scores from the final examination were adopted as the primary metric for assessing the efficacy of PLE. Data was collected from the students of Class A, where PLE was implemented in the academic year 2023, and from Classes B through E, where PLE was not implemented. The gathered data encompassed not only from the year 2023 but also from the preceding years 2022 and 2021, where PLE had not been introduced, serving as a comparative benchmark. The classes were divided into five groups based on students' enrollment numbers assigned at the time of admission. Each year, instructors are randomly allocated to one of these five divided classes. For the purpose of this study, we have designated class names as A, B, …, E corresponding to each assigned instructor.

3.5 Data Analysis

Based on the collected data, a t-test analysis was performed to compare the academic performance of students in Class A of 2023 with those in Classes B to E. Similarly, differences in grades between 2023 and the preceding years (2022 and 2021), when PLE was not implemented, were also analyzed using t-tests. This analysis aimed to evaluate the impact of introducing PLE on students' academic performance. The exercise problems and final exam questions are slightly modified each year, but there is no alteration in their level of difficulty. It should be noted in this paper that we do not compare scores or other metrics between different academic years. In 2023, 2022, and 2021, each year was divided into five classes: A, B, C, D, and E, with the same teachers assigned to each class throughout these years. However, Class B was the exception in 2023, as it was overseen by a new teacher. (Table 2)

Table 2: Classes A, B, C, D, E Assigned to teachers u, v, w, x, y, z in Each Year

	u	v	w	x	y	z
2023	A	B	C	D	E	-
2022	A	B	C	D	E	-
2021	A	-	C	D	D	B

4 Result

4.1 Summary of Results

This study compared the final exam scores of Class A students, who were introduced to Pair Learning Experiment (PLE), with those of students in Classes B to E, where PLE was not implemented. The comparison of scores was based on average marks, and the statistical significance of any differences was examined. The performance of students in Class A (who participated in PLE) versus those in Classes B, C, D, and E (who did not) was evaluated using a t-test, and statistical significance was confirmed based on the p-values between the two groups.

When the t-test was conducted for the academic year 2023, only Class D showed no statistically significant difference. Consequently, potential differences in instructor capabilities were considered, leading to a similar t-test being conducted for Classes A, B, C, D, and E in the years 2022 and 2021. These results also indicated no statistically significant differences across all the metrics, suggesting that instructor capability did not significantly influence the outcomes.

4.2 Grade Trends Analysis from 2021 to 2023

From 2021 to 2023, we compiled and analyzed the results of the final examinations for the same course. The histogram (Figure 2) analysis revealed that in 2021, there were high frequencies around 70 and 90 points, indicating two peaks. In contrast, the data for 2022 and 2023 showed a concentration of high frequencies in a single area, with a notable decline in the mode to around 70 points in 2023. Although there were slight changes in the content of the final exam questions over these three years, the overall difficulty level was consistently maintained. In the histograms, the lighter bars represent the frequencies of students in the B, C, D, and E grades, while the darker bars indicate the frequencies of students in the A grade.

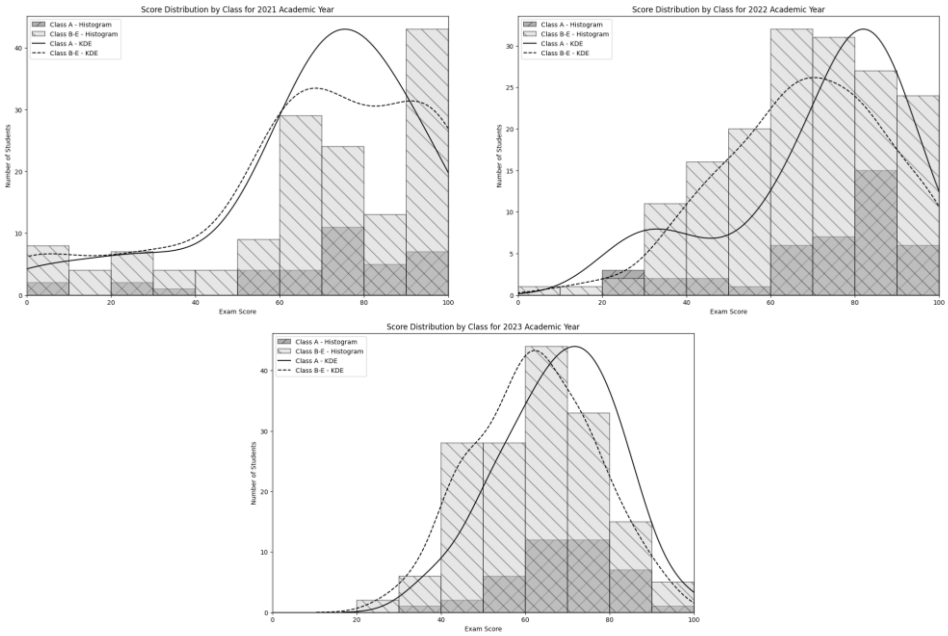
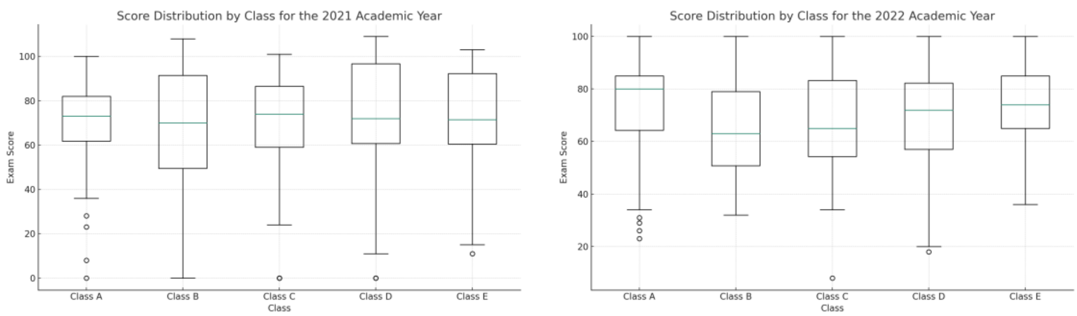


Figure 2: Histograms of Class A and Classes B, C, D, E from 2021 to 2023

The analysis of class performance from 2021 to 2023, utilizing box-and-whisker plots (Figure **3**), revealed no significant disparities in aspects such as highest, lowest, and average scores across classes. The effectiveness of PLE implementation was challenging to ascertain distinctly through the analysis via histograms and box-and-whisker plots. Consequently, a t-test was employed to statistically validate the impact of PLE.



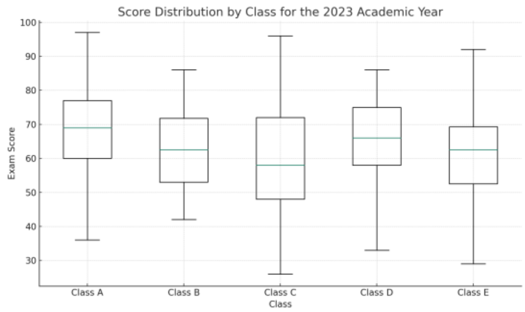


Figure 3: Box-and-Whisker Plots of Grades for Classes A, B, C, D, and E from 2021 to 2023

4.3 A vs B, C, D, E t-test

An independent two-sample t-test was conducted between Class A and the other groups, yielding t=2.29 and p=0.023. These results are statistically significant, indicating that the grades of the Class A group were superior to those of the other classes.

Further detailed analysis involved conducting t-tests between Class A and each of Classes B, C, D, and E. The t-value between Class A and Class B was 2.151, with a p-value of 0.035, while the t-value for Class A versus Class C was 2.535, with a p-value of 0.013. These outcomes confirm statistically significant differences in grades between Class A and Classes B and C. On the other hand, the t-value for Class A versus Class D was 0.871, with a p-value of 0.386, and for Class A versus Class E, it was 1.873 with a p-value of 0.065. Notably, the p-value between Class A and Class D significantly exceeded the threshold of statistical significance, and no substantial difference was recognized between Classes A and D. The p-value for Class E was also slightly above the significance level of 0.05, but not markedly different compared to that of Class D.

4.4 2023, 2022 and 2021 t-test

No statistically significant difference was found between Classes A and D in 2023. This outcome raised suspicions that the favorable performance of Class A, where PLE was implemented, might be attributed to the instructor's capabilities. Therefore, t-tests were similarly conducted for Class A versus Classes B, C, D, and E in the years 2022 and 2021, when PLE was not implemented.

The t-test results for 2022 indicated no statistically significant differences between Class A and other classes. Specifically, the t-value between Class A and Class B was 1.4448 (p=0.1524), and between Class A and Class C, it was 1.0409 (p=0.3011). Similarly, in 2021, the t-value between Class A and Class B was 0.2357 (p=0.8144), and between Class A and Class C, it was 0.7705 (p=0.4436). These findings were consistent for Classes D and E as well, with no significant differences in grades, thereby confirming that the instructional differences, including instructor capabilities in the years 2021 and 2022, did not significantly influence the outcomes.

These results suggest that the superior performance in Class A during 2023, when PLE was implemented, cannot be attributed to differences in instructor capabilities.

Table 3: T-Test Results for Scores Across 2021, 2022, and 2023: Comparative Analysis by Year (upper left:2021, upper right:2022, bottom center:2023)

Class Combination	t-Value	p-Value	Statistical Significance
A vs B	0.2357	0.8144	Not Significant
A vs C	0.7705	0.4436	Not Significant
A vs D	-0.525	0.6012	Not Significant
A vs E	-0.3419	0.7334	Not Significant

Class Combination	t-Value	p-Value	Statistical Significance
A vs B	1.4448	0.1524	Not Significant
A vs C	1.0409	0.3011	Not Significant
A vs D	0.8218	0.4136	Not Significant
A vs E	-0.203	0.8396	Not Significant

Class Combination	t-Value	p-Value	Statistical Significance
A vs B	2.1513	0.0346	Significant
A vs C	2.5351	0.0133	Significant
A vs D	0.8708	0.3865	Not Significant
A vs E	1.8727	0.0646	Significant

5 Discussion

This study aimed to evaluate the impact of introducing PLE on students' learning effectiveness and grades. The results indicated that the performance difference between Class A of 2023 and the other classes was statistically significant. Based on these findings, the following points are discussed:

Effectiveness of PLE: The introduction of PLE in Class A in 2023 led to statistically significant improvements in grades compared to other classes. This suggests that PLE effectively enhances students' learning outcomes. The mutual problem-solving and shared understanding in PLE are likely factors in improving the quality of learning. Particularly, the discussions and feedback among students could contribute to deeper understanding and new perspectives.

Influence of Instructors: While no significant difference was found between Class A and Class D in 2023, this doesn't necessarily imply that factors other than Peer Learning with Explanation (PLE), such as instructional methods or teaching materials, influenced the results. However, considering the data from 2022 and 2021, the impact of instructor capabilities appears minimal. This implies that the introduction of PLE, rather than teaching methods or materials, significantly affected the grade differences.

Comparison with Past Data: Comparing with non-PLE years (2022 and 2021), no significant differences were observed, strongly suggesting that the introduction of PLE was a key factor in improving grades. The variance in performance before and after the introduction of PLE indicates a likely impact due to PLE's implementation.

From these considerations, it seems highly plausible that the introduction of PLE contributes to enhancing students' learning effectiveness and grades. However, further research and data collection are needed to understand the effects of PLE in greater detail.

6 Conclusions

In this study, the effectiveness of Peer Learning with Explanation (PLE) in programming education was evaluated. Adapting pair programming, an agile methodology, to an educational setting, we assessed the educational impact of learners engaging with exercise problems and explaining their work and thoughts to others, as defined by the steps in PLE. Conducting experiments with first-year students from the Department of Information Media at the Faculty of Information Technology, Kanagawa Institute of Technology, we compared final examination scores between the class where PLE was implemented and those without it. The results, analyzed using a t-test, demonstrated statistically significant differences.

The methodology of PLE primarily aims to facilitate shared problem-solving processes and deepen mutual understanding among students. This study also anticipated that PLE would enhance students' comprehension and learning effects. Furthermore, the steps in PLE, ranging from explaining abstract problems to creating concrete code, suggested that this method contributes to improving students' programming skills.

However, there are some challenges in implementing PLE. These include pairing combinations, role-switching, and adapting teaching methods to account for students' abilities and experiences. Nonetheless, based on the findings of this study, it can be concluded that PLE is an effective method in programming education.

Future efforts will focus on refining the PLE methodology and applying it to a broader range of learners to further enhance its effectiveness.

Although this study identified certain outcomes regarding the effectiveness of PLE, there are aspects that require further exploration and validation.

Optimization of Pair Combinations: The pairing in this study was conducted randomly. However, combinations that consider factors like students' abilities, foundational skills, and learning styles could potentially enhance the effects of PLE. For example, pairings that complement each other's strengths and weaknesses or align learning styles might be more effective. Future research should investigate the impacts of such factor-considered pairing methods.

Continuation of Experiments: The effects of PLE were validated in the 2023 academic year classes, but the data might not be sufficient to conclude that these results will be sustained over time. Collecting data over multiple academic years and conducting long-term studies on the effects and variations of PLE will be crucial to understanding its sustained impact.

Surveying Student Attitudes and Comprehension: While the effectiveness of PLE was evaluated based on final exam grades, qualitative aspects such as students' attitudes towards PLE and their understanding of its steps were not assessed. To delve deeper into the qualitative effects of PLE and changes in student perceptions, surveys and interviews should be conducted. In particular, investigations into the quality of communication during PLE and the changes in relationships among students are warranted.

Addressing these challenges is expected to further deepen the educational effectiveness of PLE. Future research should aim to develop both the theory and practice of PLE by tackling these issues.

References

[1] Héctor Belmar, "Review on the teaching of programming and computational thinking in the world", Front. Comput. Sci., 19 October 2022, Sec. Digital Education, Volume 4 - 2022 | ; doi:10.3389/fcomp.2022.997222

[2] OECD, Digital strategies in education across OECD countries. OECD Working Papers., OECD Education Working Papers No 226, 14 September 2020

[3] Onni Aarne, Petrus Peltola, Juho Leinonen, Arto Hellas, "A Study of Pair Programming Enjoyment and Attendance using Study Motivation and Strategy Metrics", SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, February 2018, Pages 759–764; doi:10.1145/3159450.3159493

[4] Salleh, N., Mendes, E. & Grundy, J. Investigating the effects of personality traits on pair programming in a higher education setting through a family of experiments. Empir Software Eng 19, 714–752, 2014.; doi:10.1007/s10664-012-9238-4

[5] Kavitha, R. K.; Ahmed, M. S., "Knowledge Sharing through Pair Programming in Learning Environments: An Empirical Study", Education and Information Technologies, v20 n2 p319-333 Jun 2015

[6] MIYAKE, N. , "Constructive interaction and the iterative process of understanding.", Cognitive Science, 10(2): pp.151-177, 1986

[7] OKADA, T. and SIMON, H. A., "Collaborative discovery in a scientific domain." Cognitive Science, 21(2): pp.109-146, 1997

[8] SUZUKI Satoshi V., HIROKAWA Sachio, "Analysis of Learning Activity of Students in Computer Simulation Practices Introducing Pair Programming and Flip Teaching" Japan Journal of Educational Technology, 41(3) 245-253, 2018

[9] Baichang Zhong, Qiyun Wang, Jie Chen and Yi Li, "Investigating the Period of Switching Roles in Pair Programming in a Primary School", Educational Technology & Society, Vol. 20, No. 3 (July 2017), pp. 220-233

[10] Deepak Kumar, Irene Govender, "Bringing agile practice to the classroom: Student voices of third-year major project implementation", African Journal of Science, Technology, Innovation and Development, 28 October 2016

[11] Kyungsub Stephen Choi, "A comparative analysis of different gender pair combinations in pair programming", Behaviour & Information Technology, 34:8, 825-837, Published online: 24 Jul 2014.

[12] Ghobadi, S., Campbell, J. & Clegg, S., "Pair programming teams and high-quality knowledge sharing: A comparative study of coopetitive reward structures.", *Inf Syst Front* **19**, 397–409 (2017).; doi:10.1007/s10796-015-9603-0

[13] Dongo, T., Reed, A. H., & O'Hara, M. (2016). "Exploring pair programming Benefits for MIS majors" Journal of Information Technology Education: Innovations in Practice, 15, 223-239.