Conceptual Representation of the Source-Code to Support the Learning of Object-Oriented Programming Concepts

Nawras Khudhur*, Nurmaya, Yusuke Hayashi, and Tsukasa Hirashima

***Abstract*—Learning Object-Oriented Programming (OOP) concepts is a challenging task for novice learners. Previous research has examined the impact of either conceptual or practical knowledge on students' comprehension of OOP; however, there is a lack of integration between these two knowledge bases. This study proposes a novel approach that integrates theoretical and practical knowledge of OOP using a concept map to create a unified cognitive diagram that reflects the educator's understanding of the subject matter. This diagram is then used as the basis for a recomposition activity that supports student learning. An experiment is conducted in a classroom environment with 75 undergraduate grade-2 university students to investigate the effects of the proposed method. The experimental results showed that the proposed method significantly improved students' comprehension of OOP concepts, doubling their performance compared to the conventional method from pre-test to post-test. In addition, the proposed approach has the potential for early identification of low-performing students, thus allowing educators to provide targeted support to improve their learning outcomes. The integrated cognitive diagram approach proposed in this study has practical implications for educators seeking to improve the teaching of OOP concepts and identify and support low-performing students.**

***Index Terms*—Concept map, recomposition, Object-Oriented Programming (OOP) concepts, conceptual representation, theoretical knowledge, practical knowledge, concept comprehension**

I. Introduction

Computer programming is a fundamental subject in computer science-related fields and a mandatory course in many study programs. In recent decades, computer programming education has significantly shifted from the procedural programming paradigm to a more robust and productive paradigm called Object-Oriented Programming (OOP) [1]. OOP has numerous advantages, including increased coding efficiency and reusability and making problem-solving more natural and comprehensible [2].

OOP consists of several strongly interrelated concepts, including object, class, method, inheritance, polymorphism, and encapsulation [3]. Several studies have identified the difficulties students face, particularly when comprehending the relationships and interactions among OOP concepts [4–8]. Furthermore, Liberman *et al.* [9] state the difficulties in learning specific concepts and behaviors in the source-code, such as inheritance and polymorphism. It also identifies problems with the analogy approaches that are used in many classes to teach OOP concepts. They recommend a teaching model such that the instructor can identify the difficulties at an early stage. In terms of these difficulties, a lack of active practice and suitable teaching tools is considered to be one of the reasons why it is hard for students to learn about OOP concepts [10, 11].

In programming, problem-solving is a key skill that requires both theoretical and practical knowledge [12, 13]. Traditionally, students learn OOP by studying the theory of the concepts and then learning about the application of these concepts during a practical session. Previous research about computer programming for novices has mostly focused on independently supporting students in either knowledge base. However, programming in general and, specifically, the concepts of OOP are strongly related. Additionally, the interrelationship between theoretical knowledge and its application is a crucial programming aspect. Even having good programming skills without theoretical background knowledge leads to poor program design; thus, both knowledge bases are needed during the learning process [14]. Therefore, to support OOP comprehension, students need an activity that considers both knowledge bases during the learning process.

This study investigates a learning strategy combining the main aspects of OOP, namely, theoretical and practical knowledge, into one learning activity using concept maps [15]. We call this combination representation Conceptual Representation of the Source-code (CRS). In CRS, the educator creates an ideal concept map about the targeted OOP concepts and integrates it into a second concept map that implements them in source-code. The concept map is then disassembled and given to the students to be recomposed back to the original concept map. The aim of CRS is to support learners during the learning process of OOP by externalizing the OOP concepts combined with their applications and the inter- and intra-relationships rather than teaching how to write code syntactically. While the teaching of object-oriented programming has been extensively studied, the precise integration of concept mapping to bridge theoretical and practical aspects for conceptual understanding appears to be a novel contribution to the field.

In our previous pilot study, we confirmed the usability of the CRS method as a recomposition activity. In addition to the confirmed usability, the results suggested that there could be an advantage to using the CRS method in learning OOP concepts over conventional methods. However, the study lacked a control group to confirm the reliability of the learning outcome, and the properties of the CRS method were not satisfactorily conveyed to allow independent creation and

Manuscript received February 28, 2023; revised May 15, 2023; accepted July 21, 2023.

N. Khudhur, Y. Hayashi, and T. Hirashima are with the Graduate School of Advanced Science and Engineering, Hiroshima University, Hiroshima 739-8527, Japan.

Nurmaya was with Faculty of Information Technology, Universitas Yarsi, Jakarta Pusat 1051, Indonesia.

*Correspondence: nawras@lel.hiroshima-u.ac.jp (N.K.)

doi: 10.18178/ijiet.2023.13.12.1999

use by practitioners. In this study, we aim to answer the following research questions:

1) Can an educator build a practical CRS in a concept map according to the characteristics of the proposed method in an actual teaching class?
2) What is the effect of integrating the two main aspects of OOP using CRS in a concept map activity on OOP comprehension in a classroom setting compared to the conventional method?
3) How do the students perceive the proposed method's usefulness for learning OOP?

II. Background

A. Concept Map and Concept Map Recomposition

A concept map is a cognitive diagram that is used to externalize knowledge and enhance meaningful learning [15, 16]. Concept maps' visual layouts can reveal key details and conceptual relationships of the learning material to the student. In a concept map, concepts are expressed as nodes. These concepts are then connected using labels representing relationships between concepts to form a meaningful proposition. Prior studies revealed that integrating learning activities with concept maps could considerably enhance students' problem-solving skills [17–19]. In addition to promoting meaningful learning for students, using concept maps in learning could potentially improve students' interests and learning achievements better than traditional expository learning [20, 21]. Despite prior studies indicating several challenges and limitations in utilizing concept maps in academic practices for both teacher and student [22, 23], concept maps can be accepted as an alternative method for learning [24].

Essentially, there are two types of concept mapping approaches in terms of how a concept map is constructed or developed [25, 26]. The first approach offers a flexible way to develop concept maps where learners can freely integrate and customize their interests, knowledge, and external resources on a concept map. Open-ended concept mapping is often used to describe this type of concept mapping approach. The second approach is referred to as a Closed-end Concept Map (CCM). In the CCM environment, the learner receives a finite set of pre-selected concepts and links offering different cognitive activities. The CCM learning strategy helps teachers assess students' understanding and improve the quality of their teaching [27].

Hirashima *et al.* [28] introduced the Recomposition Concept Map (RCM), which adopts the CCM approach. In an RCM, domain experts or teachers create a concept map (i.e., an expert map) of a learning topic, and learners recompose the expert map by using the same set of concepts and links used in the expert map. The basic steps of an RCM activity are as follows: 1) The expert creates a concept map for a material; 2) The expert concept map is then decomposed into its basic parts by removing the connections, thus creating a kit of concepts and links; 3) The kit is then given to the learners to recompose it back to the expert map.

RCM lets the learners focus on recognizing given concepts and their relations since they are not required to create such components. Thus, it is expected to promote structural understanding [29, 30]. RCM has been confirmed to be as effective as open-end concept mapping in reading comprehension and even outperforms it in the retention task [31]. It can also be used as a formative assessment tool since it offers automatic learner map diagnosis per learner and generates an overlapped group learner map that gives insights into the aggregated learners' understanding [32]. Such aggregated comparison allows instructors to pinpoint the difficult parts of the lecture and give more accurate feedback to the learners [33]. The assessment method scores learners' concept maps by comparing each recomposed proposition to its equivalent in the expert map.

Another study by Prasteya *et al.* [34] demonstrated the effectiveness of expert map recomposition in enhancing knowledge comprehension and building upon learned knowledge. The study compared recomposed maps to open-ended concept maps and extensions. The findings revealed that students who engaged in recomposing expert maps achieved a broader and deeper understanding of the subject matter than those who started from scratch. The recomposition approach was also superior in integrating new knowledge with existing knowledge when students could freely extend their concept map, leading to improved knowledge comprehension. These results further support the utilization of expert map recomposition as a valuable strategy for promoting meaningful learning and knowledge construction in educational settings. RCM has also been proven to be suitable and significantly more efficient than traditional open-end concept mapping in collaborative environments [35]. We have adopted the recomposition concept map in this study to implement the proposed method since, in the previous study, it has been proven to be practical for teaching the concepts of OOP [36].

B. Conceptual Representation of the Source-Code (CRS)

CRS combines the two knowledge bases: the theoretical knowledge base and the practical knowledge base. Theoretical knowledge refers to the principles and concepts of a subject, while practical knowledge is the application of those concepts in actual source-code. There is a mutual and complex relationship between these two knowledge bases, and a concrete understanding of programming concepts needs to address both [14]. CRS creates an environment to support externalizing these two knowledge bases and the complex cognitive interaction between them. The objective of the CRS concept map is to visualize both knowledge bases in one diagram and act as an intermediary between the two. Bridging these two knowledge bases allows the learner to connect the concepts of OOP to their actual implementation in the source-code. Consequently, it can promote conceptual interrelationships and how concepts affect each other.

In CRS, the educator creates an ideal concept map for both theoretical and practical knowledge. The theoretical knowledge concept map describes the theory behind the concepts, like their definition, properties, how one concept relates to another concept, etc. In the practical knowledge concept map, a source-code example is prepared beforehand that applies the targeted concept(s) in practice. The structure of the source-code is then described using the concept map, such as identifying the class, methods, fields, data types, etc. Afterward, the relationship between the structure of the source-code and the theoretical background of the OOP concepts is created using appropriate links. This is a major step in CRS since these connections present to the learners how each concept is applied in the source-code. Thus, we believe that the effectiveness of the CRS method strongly depends on the suitability of the connections made between the two knowledge bases in the concept map. The guideline to create the best bridging links is not covered in this research; rather, this research investigates the effectiveness of generating a good-quality CRS in integrating theoretical and practical knowledge of OOP. The size of the concept map to describe each OOP concept is decided by the educator to fit the teaching class material and students' performance.

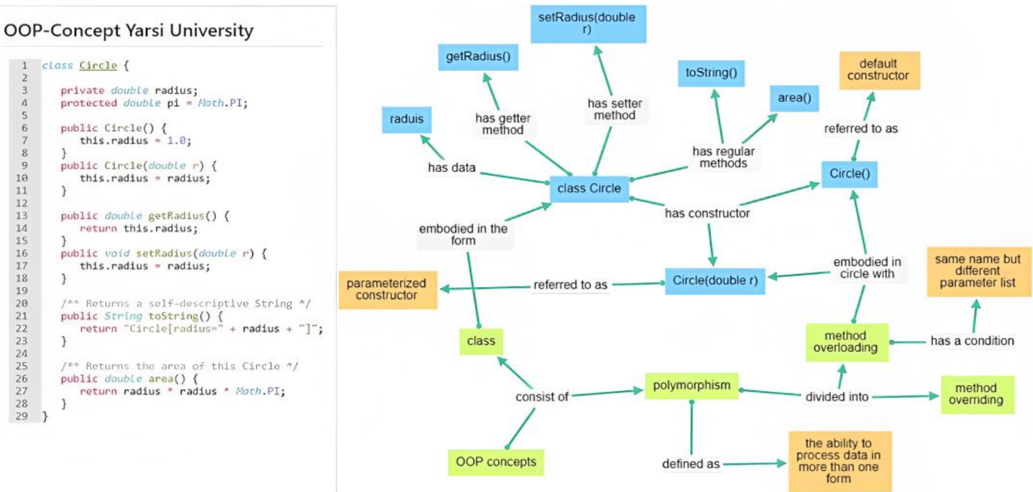

Fig. 1. CRS in concept map with a focus on the concept of overloading along with its' source-code example.

For illustration, we point to a simplified example of the CRS concept map used in the experiment in Fig. 1. By simplified, we mean that the concepts and definitions not related to the overloading type of polymorphism found in the constructor are removed from the concept map. The figure is originally in Indonesian but has been translated into English for this paper. The CRS map clarifies the theoretical background of the concepts of overloading and class, i.e., theoretical knowledge, in green. It says that the current teaching material talks about OOP concepts: class and polymorphism then give a definition or needed notation to each concept, as shown in the brown-colored concepts. The source-code is a simple example that implements constructor overloading in a class called "Circle". The structure of the source-code is drawn in the concept map, which includes all elements of the class, i.e., practical knowledge, in blue with the needed notations colored in brown. The practical knowledge concept map clarifies the methods and their types, such as regular methods, setter and getter methods, constructors, and data fields within the "Circle" class. Afterward, connecting links are added to connect both knowledge bases. These links connect the "overloading" concept to the constructors and the "class" concept to the "Circle" class in the source-code. The first link conveys the application of overloading in the source-code in the form of two constructors in class "Circle." The second link tells us that the concept of class is embodied as source-code in the "Circle" class. The concept map reveals the connection between the concepts of overloading and class.

Creating a CRS concept map continues with the educator adding relevant notations and definitions to the map as needed, based on their understanding, the needs of the target class, and their teaching approach. For instance, to visualize other abstracted concepts such as objects, an "object" of type Cylinder can be added to the CRS concept map to reveal the access restrictions of an object toward different class variables and methods. It can also be used to depict the method calls and the concepts of argument transfer and pointers. The resulting concept map and the corresponding source-code are then presented to students as a recomposition activity. The inclusion of the source-code demonstrates the application of the concepts in the targeted programming language, thus providing a clear link between the concepts and their syntax in the code. This activity is not used independently; it is used along with the corresponding material. The activity can be extended to include collaborative concept map recomposition, where students collaborate and exchange their knowledge about the OOP concepts, but the use of such collaboration is outside the scope of our paper.

III. Methodology

A comparative experiment is conducted using a between-subjects pre/post test design to address the research questions. The independent variable is the treatment practiced for OOP concept comprehension after reading the related material. In the treatment, the CRS condition was compared to the summary-making condition. Fig. 2 shows the timeline of the experiment.

The experiment flow included the following tasks:

1) Reading material about OOP concepts focusing on polymorphism and inheritance: 30 minutes.
2) Taking the comprehension test (pre-test): 30 minutes.
3) Completing the core activity depending on the condition:
   a) CRS condition: Given a source-code, recompose a concept map based on the proposed method.
   b) Summarization (SUM) condition: Given a source-code,

write a summary (about 250 words) connecting the concepts of OOP found in the material to the applied concepts in the given source-code.

4) Taking the comprehension test (post-test).
5) Answering a questionnaire about using the proposed method (CRS condition only).

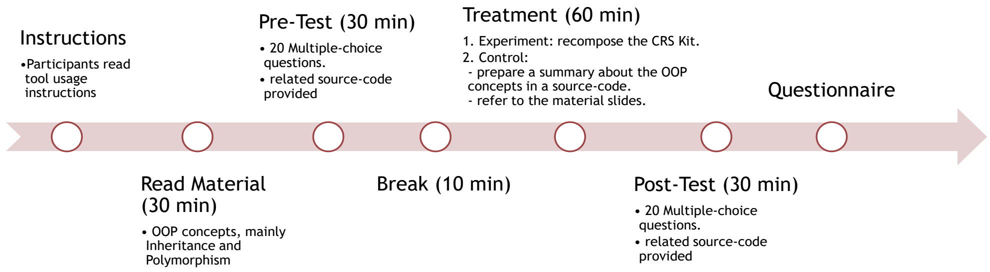

Fig. 2. Timeline of the experiment.

A. Participants

The study was conducted in an OOP class for second-year Informatics Department students at Universitas Yarsi in Indonesia. The class registrants for this class were 90 students, divided into 45 students per condition. The subject aimed to extend students' knowledge from basic programming into Object-Oriented Programming using Java. Based on that, the source-code related to the experiment was also written in the Java programming language.

B. Material

The main topics for this class material were inheritance and polymorphism in Java. The class teacher prepared digital slides explaining these topics. The class teacher regularly used the slides, regardless of the experiment. The comprehension questions[1] consisted of 20 multiple-choice items that were novel in their content but were based on the topics of the material. Each question has four options, including one correct answer. The question sequence and their corresponding options for each learner were randomly shuffled between the pre- and post-comprehension tests to reduce the chance of memorization and cheating. To ensure the suitability and effectiveness of the comprehension questions in assessing the students' integration of theoretical and practical skills, we leveraged the expertise of the class teachers, who are familiar with their classes and have several years of experience teaching the same subject. By involving the class teachers in the question preparation process, we aimed to design questions that aligned with their teaching and assessment styles. Their deep understanding of the class context and student needs allowed them to create assessments that carefully considered the integration of theoretical knowledge and practical application. Combining the teachers' familiarity with the class and our objective of assessing both theoretical and practical skills, this collaborative approach ensures a comprehensive evaluation that captures the essence of effective object-oriented programming learning.

A professional teacher with experience in OOP classes prepared the source-code in Java. The source-code consisted of two classes, "Circle" as a superclass and "Cylinder" as a subclass[2]. The source-code was reviewed and verified by the class's main teacher and the teacher assistant to be compatible with the class material and students' capabilities. The class "Circle" consists of two primitive variables, two constructors, and four methods. The class "Cylinder," on the other hand, consists of one primitive variable, two constructors, and six methods. The main class was also provided to create objects from Class "Cylinder" and invoke some methods. The source-code is used for both conditions: control and experimental. The source-code's focus is inheritance and polymorphism, which correspond to the material of the class. The inheritance concept was depicted in the Cylinder-Circle relationship. The polymorphism concept was divided into two parts: overloading and overriding. The overloading part was explained by having multiple methods with the same name but different signatures within each class of Circle and Cylinder. While the overriding concept was explained within the inheritance relationship, overriding is a phenomenon that occurs when a subclass has a method with the same name and signature as the superclass. As a result, the subclass's method will override the inherited method of the superclass. The expert map visually represents these concepts using multiple connected propositions.

The same professional teacher created the corresponding expert map, consisting of 76 propositions created with respect to the properties of CRS. The concept map is similarly reviewed and approved by the main class teacher and the assistant teacher to be used in correspondence with the class material. The expert map contained theoretical knowledge about the class material integrated with the practical knowledge represented in the source-code used in this experiment. Through review and discussions, the teachers decided on bridging links where the connection between theoretical and practical knowledge is constructed to create a high-quality CRS map. Fig. 3 shows the translated version of the final expert map created by the teachers. This satisfies our first research question about the possibility of the teacher creating a CRS concept map for an actual class by the teacher since the map has been created and approved by the teachers to be used as part of their class activity. From this expert map, the kit for recomposition was generated to be used by the experiment group. The above materials were all in Indonesian since the medium of instruction at Universitas Yarsi is Indonesian.

In addition, a 7-point Likert-type scale (1: extremely disagree to 7: extremely agree) questionnaire was prepared to

[1] https://t.ly/PU1mt
[2] https://git.io/JtKt6

measure the perceived usefulness of the proposed method. The questionnaire consists of five questions, all in Indonesian, which translate as follows: Q1. Using the CRS concept map would enable me to understand the source-code more quickly; Q2. Using the CRS concept map would enable me to understand the concepts of OOP more easily; Q3. Using the CRS concept map would enable me to understand the relationship among OOP concepts more easily; Q4. Using the CRS concept map would enhance my effectiveness in studying the programming classes; Q5. Overall, the CRS concept map is useful in this class.

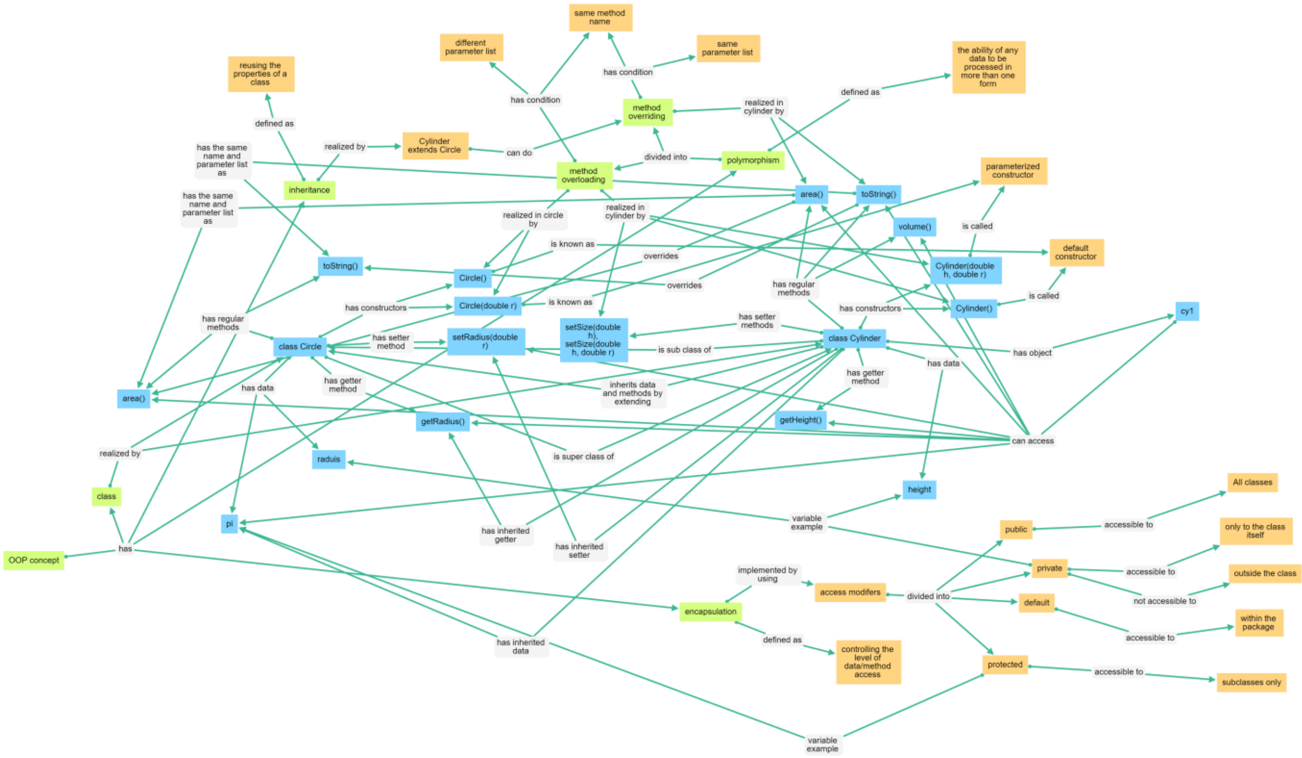

Fig. 3. Translated version of the expert map utilized in the experiment. The expert map is build based on the given source-code.

C. Procedure

A control group was needed to create a baseline for the comparison of the results of the CRS method. In the first class of the course, the class teacher administered a test to evaluate students' prior knowledge about programming. Students were divided into two groups, control and experiment (45 per condition), based on their scores on the test, such that experienced students would be distributed equally between the conditions. Such a distribution of the two conditions was later confirmed by conducting the pre-test. The experiment was conducted in the class during the regular class schedule using a tool specifically developed for the experiment. Each student had a separate account to access the tool online. The tool featured the experiment tasks in sequence according to the student's condition, with each task limited by an automatic timer. When a task was finished, the next task was automatically loaded. The students were informed about the sequence of the required tasks and the time allowed for each task. If a student did not complete a task within the specified time, the tool automatically disregarded their results. Therefore, they were advised to stay focused on the experiment and not switch to external tasks.

Following the instructions, the tool displayed the slides prepared for the class, which students could read through for 30 minutes, followed by the after-reading comprehension test (pre-test). The students were required to answer the questions by drawing on their understanding during the reading phase. Afterward, students were given 10 minutes of rest time as required by the class teacher. The break time helped refresh the students after one hour of active reading and answering the pre-test questions.

Next, the treatment phase started for 60 minutes. The CRS condition students were given the source-code and the kit of disassembled concepts and links to recompose based on the expert map. On the other hand, the SUM condition was given access to the material slides again, including the same source-code as in the CRS condition, and was required to write about 250-word summary describing the source-code in terms of the OOP concepts learned in the presentation slides.

Following the treatment phase, the participants answered the after-treatment comprehension questions (post-test), which consisted of the same questions as in the pre-test. Finally, the questionnaire about the CRS concept map recomposition experience was given to the students in the experiment condition to measure their perceived usefulness toward using this new approach to support their study of OOP.

IV. Results

From the total of 90 enrolled students, 15 were excluded from the analysis due to absences or failure to complete the experimental tasks within the allotted time. As a result, data were collected from 75 students using the provided tool, with 34 and 41 students in the control and experimental conditions, respectively. Statistical tests were conducted at α = 0.05 to address research questions. Although there are various methods for normality testing, for a small sample size ($n < 50$), the Shapiro-Wilk test is said to have more power to detect non-normality, and this is the most popular and widely used method [37]. Therefore, the Shapiro-Wilk test was used to test for the normality of the user data, and the result showed no evidence of non-normality. To confirm the homogeneity of the conditions, we used Bartlett's test of homogeneity of variances on the pre-test score. The test revealed no significant difference in variances between control and experimental conditions (Bartlett's K-squared = 0.027, df = 1, P-value = 0.87).

To answer our second research question in this experiment, "What is the effect of integrating the two main aspects of OOP using CRS in a concept map activity on OOP comprehension in an actual class setting compared to the conventional method?" we analyzed the comprehension scores of the after-reading comprehension test (pre-test) and the after-recomposition comprehension test (post-test). We performed the dependent sample t-test for the CRS condition. Table I shows the mean and standard deviation of the comprehension scores and the normalized change score for both conditions. The results revealed a significant increase in the test score from the pre-test score (M = 10.2, SD = 3.54) to the post-test score (M = 12.7, SD = 4.10); t (40) = 4.48, $P < 0.001$, with an effect size of d = 0.66, which translates to a medium-sized effect according to [38]. To understand the learning gain of the proposed method, we calculated the normalized change score and compared it against the baseline condition's normalized change score using an independent-sample t-test. The normalized change score measures the proportion of improvement or declines in academic achievement relevant to the maximum of what can be improved or lost, thus avoiding the pre-test score bias [39–41].

TABLE I: SUMMARY OF PRE- AND POST-TEST COMPREHENSION SCORES AND NORMALIZED CHANGE, ALONG WITH STATISTICAL TEST RESULTS

Group	N	Pre-test M (SD)	Post-test M (SD)	Normalized change M (SD)
SUM	34	11.1 (3.43)	12.1 (3.29)	0.12 (0.23)
CRS	41	10.2 (3.54)	12.7 (4.10)	0.26 (0.35)

Fig. 4 compares the average normalized change for both groups in a boxplot. The independent sample t-test revealed a significant difference in the normalized change score of the CRS condition (M = 0.26, SD = 0.35) compared to the control condition (M = 0.12, SD = 0.23); t (69.88) = 2.09, P = 0.039. The effect size of this analysis was d = 0.47. The effect size in terms of probability of superiority [42] reveals a 63.0% chance that a student picked at random from the CRS group will have a higher score than a student picked at random from the control group. In addition, we fitted the learning gain into the linear regression model, where the recomposition map score (M = 48.78, SD = 28.31) is used as the predictor while controlling for the pre-test score. Table II shows the results of the linear regression model. The result validates the prediction of the learning gain by the completion degree of CRS map recomposition: $R^2/R^2adj.$ = 0.218/0.176, F (2,38) = 5.285, P = 0.009.

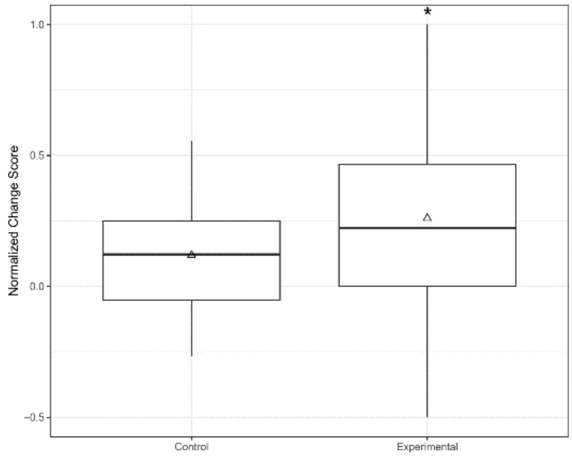

Fig. 4. A box plot comparing the normalized change score of the CRS condition to the SUM condition.

TABLE II: LINEAR REGRESSION OF LEARNING GAIN (NORMALIZED CHANGE) AS A DEPENDENT VARIABLE, AND CRS MAP SCORE WITH PRE-TEST SCORE AS INDEPENDENT VARIABLES

Predictors	Estimates	95% CI	P Value
(Intercept)	0.24	−0.07–0.55	0.124
Pre-test Score	−0.03	−0.07–0.00	0.075
CRS Map Score	0.01**	0.00–0.01	0.003
Observations		41	
R^2/R^2 adjusted		0.218/0.176	

Note: ** indicates the estimate is significant at the 0.05 level.

These results of the learning gain analysis suggest that the proposed method positively improves the learning gain of OOP concepts comprehension during the class. The improvement rate is significantly better compared to the conventional method of summarizing the concepts of OOP. The map score analysis indicates that the percentage of successfully recomposing the CRS concept map affects the learning gain positively, so it is important to focus on supporting students at the recomposition stage in future research.

The results confirm previous findings about using CRS to teach concepts of OOP [36]. However, this study confirms that the learning outcome comes from the treatment applied during the experiment. The regression analysis of the recomposition task is consistent with previous studies about using kit-build recomposition, as similar trends have been revealed where the recomposed map score played an important role in the learning activity [30, 43]. However, further research is needed to investigate the relationship between the CRS and the concept map recomposition task further.

To address the third research question, "How do the students perceive the proposed method in terms of usefulness for learning OOP?" we analyzed the subjective answers to the perceived usefulness questionnaire. The students responded to the questionnaire after completing the post-test. The inter-reliability of the questions was assessed using Cronbach's Alpha, showing good reliability with a value of α = 0.88. The summary of the answers is demonstrated in the stack chart in Fig. 5, and Table III. Each color in the figure represents one agreement scale. The number inside each color shows the number of students agreeing to the corresponding level. Regarding whether it is faster to use the proposed method to learn concepts of OOP than not, 29% of students have a neutral view, 29% have a negative view, and

41% of students agree with the improved performance. A similar percentage can be seen in questions four and five, where students were asked if the proposed method would enhance their effectiveness in studying the programming classes (46% positive, 24% negative, 30% neutral) and their overall impression of the method's usefulness (47% positive, 16% negative, 37% neutral). In questions two and three, a higher percentage of students hold positive views (59% positive, 20% negative, 20% neutral) and (53% positive, 15% negative, 32% neutral), respectively. In these two questions, the students believe that adopting the proposed method makes it easier for them to comprehend each OOP concept and the relationships among these OOP concepts. Approximately half of the students expressed a positive view of using CRS as a learning method. On the other hand, about 21% think neutrally, and less than 30% have a negative impression about using the method as a class learning tool. A high percentage of neutral and negative views can be attributed to the uniqueness of the proposed method for learning programming concepts. We believe that continuous use of the proposed method will enhance students' impressions of using CRS as a learning method in a programming class.

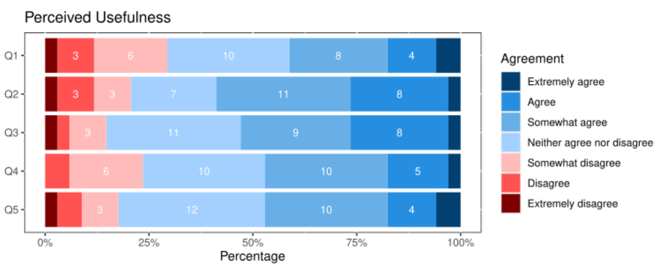

Fig. 5. Perceived usefulness questionnaire answers for each question.

TABLE III: DISTRIBUTION OF STUDENT PERCEPTIONS REGARDING THE USEFULNESS OF THE CRS METHOD FOR LEARNING OBJECT-ORIENTED PROGRAMMING CONCEPTS

Perceived Usefulness	Positive	Neutral	Negative
Question 1	41%	29%	29%
Question 2	59%	20.5%	20.5%
Question 3	53%	32%	15%
Question 4	46%	30%	24%
Question 5	47%	37%	16%
Overall Impression	49%	21%	30%

V. DISCUSSION

A. Exploratory Analysis of the Long-Term Effects of the Treatment Method

Although not part of the main research goal, the students were assessed again after one week using the same pre- and post-comprehension questions and given the same time limit for the test. In addition, their assignment and mid-term performance were tracked to answer the following exploratory research questions: 1) Compared to the conventional method, can the learning gain of CRS affect the reviewed learning gain after one week? 2) What is the correlation between performance in CRS and performance later in the course activities of assignments and mid-term tests?

The assignment score is the average score of seven consecutive weekly assignments that were part of the regular teaching plan. Each assignment was a practical code-writing activity in which students were asked to code a solution based on a given case. Assignments were submitted weekly before starting the next class, and the content was related to the material studied in the corresponding class. Similarly, the midterm score is the average of six consecutive weekly quizzes in the regular teaching plan. This formative assessment consisted of multiple-choice questions about the content of the corresponding class. The quiz was administered three days after the class in an online form. Given 15 minutes, students were asked to answer all the questions.

To investigate the first exploratory research question, we calculated the delayed normalized change from pre-test to delayed-test for both CRS ($M = 0.35$, $SD = 0.35$, $N = 39$) and SUM ($M = 0.27$, $SD = 0.30$, $N = 34$) conditions. For the delayed test, two students from the experimental condition did not make it to the class that day, so their data were excluded from the analysis. Fig. 6 shows the change in learning gain from experiment day (post) to one week later (delay). Compared to the post-normalized change score, the learning gain has increased for both conditions without any significant difference. The increase is expected, as the experiment is part of a regular class in which students have a practical session before starting the next class and participate in review studies that are likely to improve their learning.

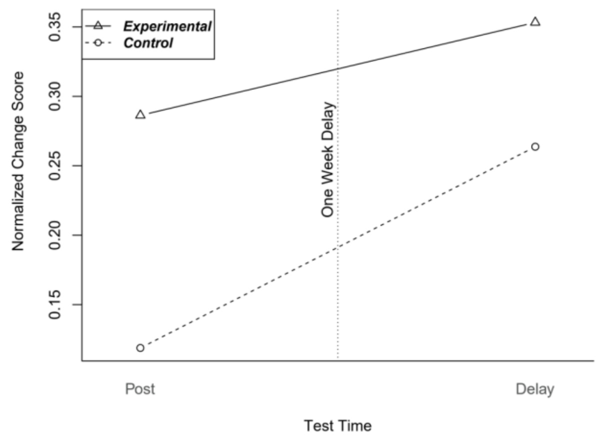

Fig. 6. The average normalized change score over a period of one week for both experiment and control conditions.

From the line chart, in addition to the learning gain increase for both conditions, we notice a higher learning slope in the control condition. Thus, we calculated the interaction between condition and time using a two-way ANOVA with Type III sums of squares, presented in Table IV. The result revealed no interaction between time and condition, meaning that continuous class improves students understanding in both conditions similarly without giving an advantage to one condition. In addition, calculating the linear regression model on the delayed normalized change revealed that for the CRS condition, their delayed learning gain is significantly predicted by their post-learning gain, R^2 / R^2 adj. = 0.319/0.281, F (2,36) = 8.435, $P < 0.001$. In contrast, the delayed learning gains of SUM condition students were not dependent on their post-learning gains.

Table V clarify the details of the linear regression model. This result potentially connects learning with the CRS method to the performance of the students during the review process, i.e., supporting the students in their review. It can indicate that if the CRS method is used continuously throughout the course, students will make greater improvements. These interpretations need further investigation, such as monitoring students' review tasks to determine how the learning gained through CRS affects their knowledge and discussion tasks during the review.

TABLE IV: ANOVA TABLE, THE RESPONSE VARIABLE IS THE LEARNING GAIN, AND THE PREDICTOR VARIABLES ARE TIME (POST/DELAY), CONDITION (EXPERIMENTAL/CONTROL), AND THEIR INTERACTION

	Df	Sum Sq	F value	Pr(>F)
group	1	0.51	5.32	0.0225*
time	1	0.36	3.73	0.0556
condition:time	1	0.06	0.58	0.449
Residuals	142	13.60		

Note: * indicates the mean difference is significant at the 0.05 level.

To investigate the second exploratory research question, "What is the correlation between CRS performance and later activity performance?" we recorded the test scores for students in the practical assignment activity and the mid-term test.

Both class activity scores were part of the regular class material given by the class teacher. Table VI shows the descriptive analysis of the class activity scores, in which the control group has a higher mean but without any significant differences. We calculated the Pearson correlation coefficient between the post-normalized change score and the assignment or mid-term test scores for both experimental and control conditions. As detailed in Table VII, the results indicate a significantly moderate to strong correlation between the assignment and mid-term test scores and the learning gain of the CRS method, respectively. In contrast, there was no significant correlation between the learning gain of the control condition and any of the class activities. This positive correlation for the CRS condition suggests that students who cannot perform well at CRS activities may also fail to perform well in later class activities; thus, it can be used to identify struggling students early in the course, thereby allowing educators to intervene and provide the necessary support to these students so they do not fall behind. Further research is needed to confirm this interpretation of the proposed method and further understand the underlying mechanisms contributing to this correlation.

TABLE V: LINEAR REGRESSION ANALYSIS OF DELAYED NORMALIZED CHANGE AS THE DEPENDENT VARIABLE, POST-NORMALIZED CHANGE AND PRE-TEST SCORE AS THE INDEPENDENT VARIABLES FOR BOTH CONDITIONS

	Predictors	Estimates	CI	p
CRS Condition	(Intercept)	0.10	−0.20–0.41	0.501
	pre-score	0.01	−0.02–0.04	0.558
	post nc	0.59	0.30–0.89	**<0.001**
	Observations		39	
	R^2/R^2 adjusted		0.319/0.281	
SUM Condition	(Intercept)	0.52	0.16–0.87	0.006
	pre-score	−0.03	−0.06–0.00	0.088
	post nc	0.25	−0.19–0.69	0.251
	Observations		34	
	R^2/R^2 adjusted		0.149/0.095	

TABLE VI: MEAN AND STANDARD DEVIATION OF THE CLASS SCORES

group	N	Assignment M (SD)	Mid-term Score M (SD)
SUM	34	64.68 (31.01)	66.43 (16)
CRS	41	61.9 (30.05)	62.17 (17.10)

TABLE VII: CORRELATION COEFFICIENT BETWEEN POST-NORMALIZED CHANGE AND CLASS ACTIVITY SCORES (ASSIGNMENT AND MID-TERM TEST)

		Assignment	Mid-term test
Norm. change score for CRS	Pearson Correlation	0.48**	0.56***
	Sig. (2-tailed)	0.0015 (adj: 0.005)	< 0.001 (adj: < 0.001)
	N	41	41
Norm. change score for SUM	Pearson Correlation	0.13	0.06
	Sig. (2-tailed)	ns	ns
	N	34	34

Note: ** and *** indicate the statistical significance of the correlation coefficient.

B. Limitations and Future Work

The key to creating an effective CRS concept map is the establishment of strong connections (bridging links) between practical and theoretical knowledge. However, we did not provide clear guidelines for achieving this integration. During the experiment, teachers achieved this through multiple rounds of discussion and revision. In future research, it is imperative to develop a framework for identifying and creating a high-quality CRS concept map with a specific focus on the "bridging links."

In the experiment, we noticed that the student's average rate of recomposition completion was not high; this suggests the difficulty of the recomposition task for the specified time period in the experiment. Therefore, it is necessary to consider the size of the CRS concept map when preparing the activity according to the ability of the students and the available time for the educator. One approach could be breaking down a big CRS concept map into smaller maps and then feeding them to the recomposition task in a scaffolding manner where the student gradually adds to the smaller CRS concept maps until reaching the full ideal map. We leave the investigation of this limitation open for other researchers. Another crucial area for future research is to assess students' misconceptions regarding OOP concepts throughout the course. In doing so, we can determine whether the CRS method is effective in preventing or rectifying these misconceptions. Additionally, examining the role of "bridging links" in this process is essential to understanding the full impact of the CRS method.

VI. CONCLUSION

In this study, we introduced a new approach to supporting OOP conceptual comprehension by integrating the two main aspects of OOP, theoretical and practical knowledge, in a concept map recomposition activity. The experimental results revealed the following interpretations: 1) An activity using the proposed method in an actual class is practical for an educator; On the other hand, students hold a welcoming view of using the proposed method as a learning tool during their study of OOP. Thus, it gives educators an effective method for conducting activities that promote conceptual comprehension of OOP concepts; 2) The proposed method positively affects OOP concept comprehension during the class. The results reveal the importance of integrating theoretical and practical knowledge for students learning OOP; 3) There is a potential for long-term usage of the proposed method since, under experimental conditions, students showed that the learning gained through the proposed method can significantly explain their review performance over one week, a standard period between classes in educational institutions; 4) Students who are good at the proposed method activity tend to also perform better in the subsequent activities of the class, including code-writing activities, and vice versa. Thus, it helps educators find low-performing students at an early stage of teaching to interfere with and support those students in their learning process.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

NK developed the software used in the experiment, analyzed the data, and wrote the manuscript. NK and NN conducted and managed the experiment; YH and TH contributed to the data analysis and discussion of the results; all authors had approved the final version.

FUNDING

This work is partially supported by JSPS KAKENHI Grant Number 19H04227.

REFERENCES

[1] M. Kolling and J. Rosenberg, "An object-oriented program development environment for the first programming course," *ACM SIGCSE Bulletin.*, vol. 28, no. 1, pp. 83–87, 1996.

[2] A. I. Isaiah, A. C. Odi, A. U. Rita, A. C. Verginia, and O. H. Anaya, "Technological advancement in object oriented programming paradigm for software development," *International Journal of Applied Engineering Research*, vol. 14, no. 8, pp. 1835–1841, 2019.

[3] D. J. Armstrong, "The quarks of object-oriented development," *Communications of the ACM*, vol. 49, no. 2, pp. 123–128, Feb. 2006.

[4] L. C. Kaczmarczyk, E. R. Petrick, J. P. East, and G. L. Herman, "Identifying student misconceptions of programming," in *Proc. the 41st ACM Technical Symposium on Computer Science Education, ser. SIGCSE '10*, New York, NY, USA: Association for Computing Machinery, 2010, pp. 107–111.

[5] N. Ragonis and M. Ben-Ari, "On understanding the statics and dynamics of object-oriented programs," in *Proc. the 36th SIGCSE Technical Symposium on Computer Science Education, ser. SIGCSE '05*, New York, NY, USA: Association for Computing Machinery, 2005, pp. 226–230.

[6] M. Teif and O. Hazzan, "Partonomy and taxonomy in object-oriented thinking: Junior high school students' perceptions of object-oriented basic concepts," in Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education, 2006, pp. 55–60.

[7] J. Sorva, "Misconceptions and the beginner programmer," *Computer Science Education: Perspectives on Teaching and Learning in School*, p. 171, Bloomsbury, 2018.

[8] J. Sajaniemi, M. Kuittinen, and T. Tikansalo, "A study of the development of students' visualizations of program state during an elementary object-oriented programming course," *Journal on Educational Resources in Computing*, vol. 7, no. 4, pp. 1–31, 2008.

[9] N. Liberman, C. Beeri, and Y. Ben-David Kolikant, "Difficulties in learning inheritance and polymorphism," *ACM Transactions on Computing Education*, vol. 11, no. 1, pp. 1–23, Feb. 2011, https://doi.org/10.1145/1921607.1921611

[10] K. A.-M. Sarpong, J. K. Arthur, and P. Y. O. Amoako, "Causes of failure of students in computer programming courses: The teacher-learner perspective," *International Journal of Computer Applications*, vol. 77, no. 12, pp. 27–32, 2013.

[11] M. Kolling, B. Quig, A. Patterson, and J. Rosenberg, "The BlueJ system and its pedagogy," *Computer Science Education*, vol. 13, no. 4, pp. 249–268, 2003.

[12] A. K. Whitfield, S. Blakeway, G. E. Herterich, and C. Beaumont, "Programming, disciplines and methods adopted at Liverpool Hope University," *Innovation in Teaching and Learning in Information and Computer Sciences*, vol. 6, no. 4, pp. 145–168, 2007.

[13] M. Wang, B. Wu, P. A. Kirschner, and J. M. Spector, "Using cognitive mapping to foster deeper learning with complex problems in a computer-based environment," *Computers in Human Behavior*, vol. 87, pp. 450–458, 2018.

[14] A. Eckerdal, "Novice programming students' learning of concepts and practise," Ph.D. dissertation, Uppsala University, Division of Scientific Computing, Numerical Analysis, 2009.

[15] J. D. Novak, "Results and implications of a 12-year longitudinal study of science concept learning," *Research in Science Education*, vol. 35, no. 1, pp. 23–40, Mar 2005.

[16] Y. Rosen and M. Tager, "Making student thinking visible through a concept map in computer-based assessment of critical thinking," *Journal of Educational Computing Research*, vol. 50, no. 2, pp. 249–270, 2014.

[17] G.-J. Hwang, F.-R. Kuo, N.-S. Chen, and H.-J. Ho, "Effects of an integrated concept mapping and web-based problem-solving approach on students' learning achievements, perceptions and cognitive loads," *Computers & Education*, vol. 71, pp. 77–86, 2014, https://doi.org/10.1016/j.compedu.2013.09.013

[18] S. Kamble and B. Tembe, "The effect of use of concept maps on problem solving performance and attitude in mechanical engineering course," *Procedia-Social and Behavioral Sciences*, vol. 83, pp. 748–754, 2013.

[19] M. Karadag, "The effect of concept map based education on the problem solving skills of students," *New Trends and Issues Proceedings on Humanities and Social Sciences*, vol. 3, no. 1, pp. 506–513, 2017.

[20] G.-J. Hwang, L.-H. Yang, and S.-Y. Wang, "A concept map-embedded educational computer game for improving students' learning performance in natural science courses," *Computers & Education*, vol. 69, pp. 121–130, 2013.

[21] C.-C. Chiou, "The effect of concept mapping on students' learning achievements and interests," *Innovations in Education and Teaching International*, vol. 45, pp. 375–387, 2008.

[22] A. Pinandito, D. D. Prasetya, Y. Hayashi, and T. Hirashima, "Design and development of semi-automatic concept map authoring support tool," *Research and Practice in Technology Enhanced Learning*, vol. 16, no. 1, p. 8, 2021.

[23] D. O. Ashipala, S. Elias, and A. Lifalaza, "Nursing students' experiences of utilizing a concept map as a learning method in human anatomy and physiology: A qualitative descriptive study," *International Journal of Africa Nursing Sciences*, vol. 18, 100547, 2023.

[24] C. T. Machado and A. A. Carvalho, "Concept mapping: Benefits and challenges in higher education," *The Journal of Continuing Higher Education*, vol. 68, no. 1, pp. 38–53, 2020.

[25] K. Oliver, "A comparison of web-based concept mapping tasks for alter- native assessment in distance teacher education," *Journal of Computing in Teacher Education*, vol. 24, no. 3, pp. 95–103, 2008.

[26] P. G. F. Furtado, T. Hirashima, and Y. Hayashi, "Reducing cognitive load during closed concept map construction and consequences on reading comprehension and retention," *IEEE Transactions on Learning Technologies*, vol. 12, no. 3, pp. 402–412, 2019, https://doi.org/10.1109/TLT.2018.2861744

[27] P.-H. Wu, G.-J. Hwang, M. Milrad, H.-R. Ke, and Y.-M. Huang, "An innovative concept map approach for improving students' learning performance with an instant feedback mechanism," *British Journal of Educational Technology*, vol. 43, no. 2, pp. 217–232, 2012.

[28] T. Hirashima, K. Yamasaki, H. Fukuda, and H. Funaoi, "Framework of kit-build concept map for automatic diagnosis and its preliminary use," *Research and Practice in Technology Enhanced Learning*, vol. 10, no. 1, p. 17, 2015.

[29] T. Hirashima, "Reconstructional concept map: automatic assessment and reciprocal reconstruction," *International Journal of Innovation, Creativity and Change*, vol. 5, pp. 669–682, 2019.

[30] B. S. Andoko, Y. Hayashi, T. Hirashima, and A. N. Asri, "Improving English reading for EFL readers with reviewing kit-build concept map," *Research and Practice in Technology Enhanced Learning*, vol. 15, no. 1, p. 7, 2020.

[31] M. Alkhateeb, Y. Hayashi, T. Rajab, and T. Hirashima, "Comparison between kit-build and scratch-build concept mapping methods in supporting EFL reading comprehension," *The Journal of Information and Systems in Education*, vol. 14, no. 1, pp. 13–27, 2015.

[32] J. Pailai, W. Wunnasri, K. Yoshida, Y. Hayashi, and T. Hirashima, "The practical use of kit-build concept map on formative assessment," *Research and Practice in Technology Enhanced Learning*, vol. 12, no. 1, pp. 1–23, 2017.

[33] K. Sugihara, T. Osada, S. Nakata, H. Funaoi, and T. Hirashima, "Experimental evaluation of kit-build concept map for science classes in an elementary school," in *Proc. ICCE2012*, pp. 17–24, 2012.

[34] D. D. Prasetya, T. Hirashima, and Y. Hayashi, "Comparing two extended concept mapping approaches to investigate the distribution of students' achievements," *IEICE Transactions on Information and Systems*, vol. E104.D, no. 2, pp. 337–340, Feb. 2021.

[35] A. Pinandito, C. P. Wulandari, D. D. Prasetya, N. Khudhur, Y. Hayashi, and T. Hirashima, "Efficient online collaborative learning through concept mapping with kit-build concept map," in *Proc. 2021 the 6th International Conference on Sustainable Information Engineering and Technology*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 125–131.

[36] N. Khudhur, P. G. F. Furtado, A. Pinandito, S. Matsumoto, Y. Hayashi, and T. Hirashima, "Conceptual level comprehension support of the object-oriented programming source-code using kit-build concept map," in *Proc. 29th International Conference on Computers in Education, ICCE 2021*, 2021, vol. 1, pp. 315–320.

[37] P. Mishra, C. M. Pandey, U. Singh, A. Gupta, C. Sahu, and A. Keshri, "Descriptive statistics and normality tests for statistical data," *Annals of Cardiac Anaesthesia*, vol. 22, no. 1, pp. 67–72, 2019.

[38] P. A. Lachenbruch. "Statistical power analysis for the behavioral sciences," *Journal of the American Statistical Association*, vol. 84, no. 408, pp. 1096–1097, 1989.

[39] J. D. Marx and K. Cummings, "Normalized change," *American Journal of Physics*, vol. 75, no. 1, pp. 87–91, 2007.

[40] S. Perdana and D. Azhari, "Addressing an undergraduate research issue about normalized change for critical thinking test," *Jurnal Pendidikan IPA Indonesia*, vol. 6, 2017.

[41] A. R. Setiawan and M. Kudus, "What is the best way to analyze pre–post data," EdArXiv Preprints, 2020, https://doi.org/10.35542/osf.io/h4e6q

[42] R. J. Grissom and J. J. Kim, *Effect Sizes for Research: A Broad Practical Approach*, Lawrence Erlbaum Associates Publishers, 2005.

[43] W. Wunnasri, J. Pailai, Y. Hayashi, and T. Hirashima, "Reliability investigation of automatic assessment of learner-build concept map with kit-build method by comparing with manual methods," in *Proc. International Conference on Artificial Intelligence in Education*, Springer, 2017, pp. 418–429.