# Investigation of Value Trace Problem for C++ Programming Self-study of Novice Students

Shune Lae Aung, Nem Khan Dim, Soe Mya Mya Aye, Nobuo Funabiki, and Htoo Htoo Sandi Kyaw

***Abstract*—Presently, C++ programming has been taught in many universities around the world as the first object-oriented programming language for undergraduate students to start studying programming concepts and computer architecture. However, many of them may struggle in studying C++ programming due to the nature in the formal language. Thus, hands-on self-study tools can be effective. In previous, we have developed Java programming learning assistant system (JPLAS) for assisting self-study of Java programming. JPLAS offers the value trace problem (VTP) for novice students to study Java programming through code reading study. In a VTP instance, actual values of important variable or standard output messages in a given source code are questioned, where the correctness of each answer is checked through string matching. In this paper, we investigate the effectiveness of VTP for hands-on self-study of C++ programming by novice students. We generated 37 VTP instances for basic grammar concepts using source codes in textbooks or websites for C++ programming, and asked 46 students in three universities in three countries to solve them using the answer interface for JPLAS. The results suggest that most of the students are satisfactory, but some students need cares at early programming study stage.**

***Index Terms*—C++ programming, code reading, grammar concepts, value trace problem.**

## I. Introduction

The programming language C++ has been popular even now as a general purpose language that has been designed to making programs in the variety of application domains. According to the IEEE Spectrum [1], C++ is currently the fourth popular programming language, even though long time has passed since the appearance. C++ is an extension of the traditional programming language C, to offer the advantages of the object oriented programming. C++ keeps the functional features of C to facilitate the low level manipulations of computers by programs. As a result, C++ programming is broadly taught as a fundamental programming language in many universities around the world.

From viewpoints of programming educations, C++ can be a good start for studying programming from the ground up if compared with more advanced and practical ones such as Java, Python, and Ruby. C++ programming supports accessing memories and registers in computers by programs for low level manipulations, which is also useful in studying the computer architecture. At the same time, it allows the object oriented programming features as the more advanced and abstract programming. C++ is actually used in almost every area of software developments, such as application software, programming languages, game developments, and embedded systems. However, many students are still struggling in studying C++ due to the nature in the formal language. Hands-on self-study tools can be effective to solve it.

In previous, we have developed Java programming learning assistant system (JPLAS) for assisting self-study of Java programming [2]. JPLAS provides several types of *exercise problems* to assist *Java programming* studies at various levels. One of them is the *value trace problem (VTP) that* is designed for novice students to study basic grammar concepts and programming skills through *code reading* [3]. A VTP instance consists of a source code, a set of questions, and correct answer strings. Each question requests to answer the actual value of an important variable or a standard output message in the source code. The correctness of any answer from a student is marked through *string matching* with the stored correct answer string in the web-based offline *answer interface* [4].

In this paper, we investigate the effectiveness of the *value trace problem (VTP)* for hands-on self-study of *C++ programming* by novice students, based on the works for Java programming. For the effective VTP instances for C++ programming study, it is important to select proper source codes that contain basic grammar concepts to be studied by the students. It is emphasized that C++ greatly differs from Java in the memory managements, such as allocating and de allocating memory manually by the program. C++ supports the memory access using a *pointer* for implementing fast and efficient programs. These concepts must be covered in the generated VTP instances to understand the advantage in C and C++ programming. In addition, the object-oriented programming concept of C++ programming must be studied.

Based on the abovementioned conditions, we collected 37 source codes from websites and textbooks for basic grammar concepts in C++ programming, and generated VTP instances manually, after analyzing important variables and outputs messages in the codes.

Then, to verify the effectiveness of the generated 37 VTP instances, we assigned them to a total of 46 students in three universities in three countries. They include 17 first-year or second-year undergraduate students in Yangon University in Myanmar, 13 graduate students in Okayama University in Japan, and 16 graduate students in Electronic Engineering

Manuscript received January 11, 2022; revised March 9, 2022.

S. L. Aung and N. Funabiki are with Okayama University, Okayama, Japan (e-mail: p1pp6ypa@s.okayama-u.ac.jp, funabiki@okayama-u.ac.jp).

N. K. Dim and S. M. M. Aye are with Yangon University, Yangon, Myanmar (e-mail: nemkdim@gmail.com, smyamyaaye@gmail.com).

H. H. S. Kyaw is with Tokyo University of Agriculture and Technology, Tokyo, Japan (e-mail: htoohtoosk@go.tuat.ac.jp).

doi: 10.18178/ijiet.2022.12.7.1663

Polytechnic Institute of Surabaya University in Indonesia. They are either currently studying or have studied C++ programming in universities. These students were requested to solve them at home using the answer interface for JPLAS by themselves as self-study without tutoring from the teacher. Their solution results show that among the 46 students, 35 students solved all the questions correctly. Only two students had the difficulty in solving them. These students need cares from the teacher at this early programming study stage.

The rest of this paper is organized as follows: Section II review preliminary works related to this study and literature. Section III presents method for value trace problems for C++ programming. Section IV includes the result and discussion based on evaluating them through applications to students. Finally, Section V concludes this paper with future work.

## II. Preliminary Work

In this section, we review our preliminary works to this study. We also discuss some related works in literature.

### A. Answer Interface

To solve exercise problems in JPLAS without the Internet connection at home, the web-based *offline answer interface* has been implemented in [4]. This function supports students to solve VTP instances on a browser, to keep the compatibility with the online JPLAS [5]. The answer marking procedure by string matching is implemented using JavaScript to run on the browser. The function contains the correct answers of the questions, where they are encrypted using *SHA256*.

### B. VTP Generation Procedure

To compose a VTP instance, a teacher needs to prepare a *source code*, a set of *questions*, the *answer forms* (blanks), and the *correct answers*. To generate a new VTP instance, the following procedure should be complied:

1) Select a source code to be studied by the students from a website or a textbook.
2) Find the important variables and standard output messages in the source code, and make the questions of asking the actual values or messages of them.
3) Insert the standard output statements into the source code to output them by compiling and running it.
4) Collect the correct answers to the questions by running the code and observing the corresponding standard outputs from the code.
5) Make the source code, the questions, and the correct answers into one text file.
6) Run the program in [4] with the text file in e), and generate the HTML/CSS/JavaScript files for the offline answer interface for this VTP instance.
7) Register the generated VTP instance in the assignment to students.

### C. Related Works in Literature

Here, we briefly discuss related works in literature as follow.

In [6], Hwang *et al*. proposed the web-based programming assisted system for cooperation (WPASC) designing learning activity for facilitating cooperative programming learning, and investigated cooperative programming learning behaviors of students and the relationships with learning performances.

In [7], Quinson *et al*. presented the programmer's learning machine (PLM) as an interactive exerciser aimed at learning programming and algorithms. It targets students in (semi) autonomous settings, using an integrated and graphical environment that provides a short feedback loop. This generic platform also enables teachers to create specific programming micro-worlds that match their teaching goals. PLM provides two main panels to provide information for students to solve exercises.

In [8], Samy *et al*. developed the intelligent tutoring system called the CPP tutor for the interactive learning environment for students. Students will be able to learn C++ programming and to gain knowledge more quickly and effectively than students using traditional methods of teaching.

In [9], Ihantola *et al*. presented a systematic literature review for automated assessments of programming assignments, which includes major features and approaches from the pedagogical and technical points of view.

In [10], Jain *et al*. developed an educational tool for understanding algorithm, and building and learning programming language. This tool provides an innovative and unified graphical user interface for developments of multimedia objects, educational games, and applications. It also provides an innovative method for code generations to enable students to learn the basics of programming languages using drag-n-drop methods of image objects.

In this paper, we generate value trace problems for C++ programming and investigate the effectiveness for novice students to learn basic grammar concepts through code reading by themselves. Students can get learning achievement by reading and understanding the code by using offline answer interface.

## III. Method for Value Trace Problem for C++ Programming

In this section, we present generations of value trace problems (VTPs) for studying basic grammar concepts in C++ programming.

### A. VTP for Basic Grammar Concept

In this paper, we generated 37 VTPs using source codes in websites [11]-[13] and textbook [14] for basic grammar concepts in C++ programming. Table I shows the instance ID, the basic grammar concept, the number of lines in the source code, the number of questions, and the number of answer forms on each VTP instance.

### B. Example VTP Instance

Fig. 1 illustrates the answer interface for the example VTP instance at ID=15. This question asks the values of important variables, *i* and *cars[i]*. Their correct answers are 0, Volvo, 1, BMW, 2, Ford, 3, and Mazda.

A student needs to read the source code carefully to understand it, fill in the forms, and click the "Answer" button.

Then, the form becomes white if the answer is correct, and red otherwise. The student can repeat the answering process until all the answers become correct.

TABLE I: VTP INSTNACES FOR BASIC GRAMMAR CONCEPTS

| ID | Basic grammar concepts | # of lines | # of questions | # of forms |
|---|---|---|---|---|
| 1 | variable usage | 10 | 3 | 3 |
| 2 | data type, size usage | 12 | 6 | 6 |
| 3 | arithmetic operators | 14 | 4 | 4 |
| 4 | relational operators | 14 | 3 | 5 |
| 5 | if statement | 19 | 2 | 2 |
| 6 | if else statement | 19 | 2 | 2 |
| 7 | nested if statement | 22 | 2 | 2 |
| 8 | switch statement | 31 | 2 | 2 |
| 9 | default value in switch statement | 18 | 2 | 2 |
| 10 | nested switch statement | 20 | 4 | 4 |
| 11 | for loop | 11 | 1 | 1 |
| | while loop | 12 | 5 | 5 |
| 13 | do while loop | 14 | 5 | 5 |
| 14 | nested loop | 13 | 1 | 18 |
| 15 | one dimensional array | 12 | 1 | 18 |
| 16 | finding average of a set of values | 24 | 2 | 2 |
| 17 | two dimensional array | 16 | 6 | 6 |
| 18 | string copy function | 13 | 2 | 2 |
| 19 | string length function | 9 | 1 | 1 |
| 20 | combining string and number | 13 | 3 | 3 |
| 21 | pointer | 11 | 2 | 2 |
| 22 | reverse case using array indexing | 18 | 2 | 2 |
| 23 | function | 14 | 2 | 2 |
| 24 | function with parameters | 14 | 3 | 3 |
| 25 | function adding two numbers | 13 | 1 | 1 |
| 26 | local variable | 14 | 2 | 2 |
| 27 | global variable | 16 | 2 | 2 |
| 28 | return value in function | 11 | 1 | 1 |
| 29 | function with call by reference | 19 | 2 | 4 |
| 30 | function overloading | 25 | 3 | 4 |
| 31 | class and object | 22 | 4 | 4 |
| 32 | method in class | 20 | 1 | 2 |
| 33 | constructor | 26 | 6 | 6 |
| 34 | encapsulation | 29 | 1 | 1 |
| 35 | polymorphism | 37 | 1 | 3 |
| 36 | Inheritance | 24 | 1 | 2 |
| 37 | exception handling | 22 | 2 | 2 |
| | average | 17.6 | 2.5 | 3.4 |
| | total | 651 | 93 | 126 |

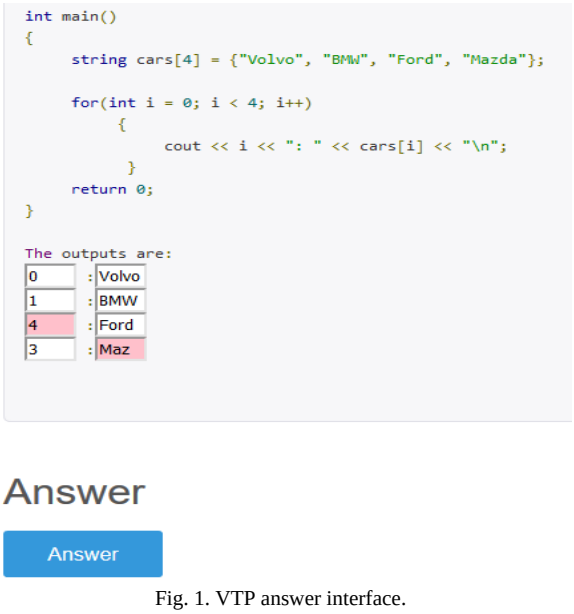


Fig. 1. VTP answer interface.

## IV. RESULT AND DISCUSSION

In this section, we evaluate the generated 37 VTP instances for C++ programming through applications to a total of 46 students in three universities in three countries, namely, 17 first-year or second-year undergraduate students in Yangon University, Myanmar, 13 graduate students in Okayama University, Japan, and 16 graduate students in Electronic Engineering Polytechnic Institute of Surabaya University, Indonesia. We analyze their solution results.

### A. VTP for Basic Grammar Concept

As shown in Table I, the 37 instances contain the total of 126 answer forms (blanks). We asked the students to solve them using the offline answer interface for JPLAS at home. These students have studied basic grammar concepts for C++ programming. The first-year and second-year undergraduate students in Yangon University have studied them for three months, the second students have for at least one year, and the graduate students have for several months at their undergraduate programs.

### B. Student Solution Results

First, we analyze the performance of each student in solving the VTP instances. Table II shows the student ID, the number of correctly solved answer forms, the total number of answer submission times to mark the answers, the missing instance ID in solving, and the correct answer rate among the 126 forms.

TABLE II: RESULT FOR EACH STUDENT

| Student ID | # of correct answers | # of submissions | # of missing instances | Correct answer rate |
|---|---|---|---|---|
| 1 | 126 | 81 | 0 | 100% |
| 2 | 126 | 50 | 0 | 100% |
| 3 | 126 | 37 | 0 | 100% |
| 4 | 126 | 63 | 0 | 100% |
| 5 | 120 | 61 | 0 | 91% |
| 6 | 122 | 38 | 0 | 98% |
| 7 | 126 | 37 | 0 | 100% |
| 8 | 126 | 52 | 0 | 100% |
| 9 | 126 | 54 | 0 | 100% |
| 10 | 126 | 40 | 0 | 100% |
| 11 | 126 | 50 | 0 | 100% |
| 12 | 126 | 92 | 0 | 100% |
| 13 | 126 | 59 | 0 | 100% |
| 14 | 107 | 39 | 0 | 81% |
| 15 | 109 | 42 | 0 | 85% |
| 16 | 121 | 56 | 20 | 97% |
| 17 | 121 | 43 | 18 | 96% |
| 18 | 126 | 43 | 0 | 100% |
| 19 | 126 | 37 | 0 | 100% |
| 20 | 126 | 50 | 0 | 100% |
| 21 | 126 | 55 | 0 | 100% |
| 22 | 126 | 40 | 0 | 100% |
| 23 | 125 | 53 | 0 | 99% |
| 24 | 125 | 87 | 0 | 99% |
| 25 | 126 | 53 | 0 | 100% |
| 26 | 126 | 84 | 0 | 100% |
| 27 | 116 | 65 | 0 | 92% |
| 28 | 126 | 87 | 0 | 100% |
| 29 | 122 | 55 | 0 | 97% |
| 30 | 122 | 55 | 0 | 97% |
| 31 | 126 | 96 | 0 | 100% |
| 32 | 126 | 43 | 0 | 100% |
| 33 | 126 | 74 | 0 | 100% |
| 34 | 126 | 54 | 0 | 100% |
| 35 | 126 | 58 | 0 | 100% |
| 36 | 126 | 43 | 0 | 100% |

| | | | | |
|---|---|---|---|---|
| 37 | 126 | 62 | 0 | 100% |
| 38 | 126 | 46 | 0 | 100% |
| 39 | 126 | 72 | 0 | 100% |
| 40 | 126 | 61 | 0 | 100% |
| 41 | 126 | 48 | 0 | 100% |
| 42 | 126 | 41 | 0 | 100% |
| 43 | 126 | 88 | 0 | 100% |
| 44 | 126 | 80 | 0 | 100% |
| 45 | 126 | 78 | 0 | 100% |
| 46 | 126 | 69 | 0 | 100% |
| average | 124.3 | 58.1 | 0.8 | 98.5% |
| SD | 4.1 | 16.8 | 3.9 | 0.00 |

*1) Individual student analysis*

Based on the results, most of the students solved all the instances correctly. Particularly, the three students with ID=3, 7, and 19 are excellent, because they solved every VTP instance with only one submission. On the other hand, the 11 students with ID=5, 6, 14, 15, 16, 17, 23, 24, 27, 29 and 30 could not solve some instances. The two students with ID=16 and ID=17 missed to solve one instance, although they show the high answer rate 97% and 96% respectively. Although they should be more careful in solving all the instances, it will be necessary to improve the interface of the answer interface to avoid it. The two students with ID=14 and 15 show the lower correct rate than 90%. These students will need a lot of efforts to improve the understanding of C++ programming.

The average number of submission times was 58.1, which suggests that a student submitted his/her answers for one instance two times on average. The standard deviation (SD) is 16.8, which suggests the large diversity among the students. The average correct answer rate 98.5% indicates that a student correctly solved 98.5% of the questions on average.

*2) Correct answer rate distribution*

Table III shows the distribution of the correct answer rates of the students. This table indicates that 35 students among 46 achieved the 100% rate, and nine did over 90% correct rate. Only two students did under 90% rate, who will need to take time for learning the fundamental C++ programming.

TABLE III: CORRECT ANSWER RATE DISTRIBUTION OF STUDENTS

| Range of correct answer rate | # of students |
|---|---|
| 80% - 84% | 1 |
| 85% - 89% | 1 |
| 90% - 94% | 2 |
| 95% - 99% | 7 |
| 100% | 35 |

*3) Submission time distribution*

Table IV shows the distribution of the numbers of answer submission times by the students. The three students with ID=3, 7, and 19 correctly solved every instance by submitting the answer only once. They are excellent students and understand *C++ programming* very well. The nine students with ID=1, 12, 24, 26, 28, 31, 43, 44, and 45 submitted the answers more than 75 times. Among them, the eight students with ID=1, 12, 26, 28, 31, 43, 44, and 45 achieved the over 100% correct rate. Thus, these students were seriously solving the VTP instances by submitting answers many times, and actually reached the correct answers, although they will need more practices in *C++ programming*.

TABLE IV: SUBMISSIONS TIMES DISTRIBUTION OF STUDENTS

| Submission time range | # of students |
|---|---|
| 37 | 3 |
| 38 – 49 | 12 |
| 50 – 74 | 22 |
| 75 – 99 | 9 |

## C. Individual VTP Instance Result

Next, we analyze the solving results of the individual VTP instances by the students. Table V shows the instance ID, the number of students who did not attempt to solve, the total number of answer submissions to mark the answers, and the average correct answer rate among the 46 students.

TABLE V: RESULT FOR EACH VTP INSTANCE

| Instance ID | # of unattempted students | # of submissions | Average correct rate |
|---|---|---|---|
| 1 | 0 | 72 | 100% |
| 2 | 0 | 122 | 99% |
| 3 | 0 | 58 | 100% |
| 4 | 0 | 96 | 100% |
| 5 | 0 | 70 | 100% |
| 6 | 0 | 51 | 100% |
| 7 | 0 | 50 | 100% |
| 8 | 0 | 55 | 100% |
| 9 | 0 | 55 | 100% |
| 10 | 0 | 66 | 100% |
| 11 | 0 | 87 | 100% |
| 12 | 0 | 55 | 100% |
| 13 | 0 | 67 | 100% |
| 14 | 0 | 65 | 100% |
| 15 | 0 | 56 | 100% |
| 16 | 0 | 113 | 98% |
| 17 | 0 | 70 | 94% |
| 18 | 1 | 103 | 99% |
| 19 | 0 | 54 | 100% |
| 20 | 1 | 72 | 99% |
| 21 | 0 | 70 | 96% |
| 22 | 0 | 86 | 100% |
| 23 | 0 | 53 | 100% |
| 24 | 0 | 71 | 100% |
| 25 | 0 | 50 | 100% |
| 26 | 0 | 51 | 100% |
| 27 | 0 | 70 | 100% |
| 28 | 0 | 54 | 100% |
| 29 | 0 | 51 | 100% |
| 30 | 0 | 50 | 100% |
| 31 | 0 | 69 | 96% |
| 32 | 0 | 82 | 94% |
| 33 | 0 | 72 | 100% |
| 34 | 0 | 53 | 96% |
| 35 | 0 | 69 | 96% |
| 36 | 0 | 215 | 88% |
| 37 | 0 | 68 | 100% |
| average | 0.054 | 72 | 99% |
| SD | 0.229 | 29.9 | 0.03 |

*1) Individual instance analysis*

Table V indicates that in general, the 37 VTP instances are relatively easy for them. The average correct answer rate is 100% for any instance except for the 11 instances with ID=2, 16, 17, 18, 20, 21, 31, 32, 34, 35, and 36. These instances may be relatively difficult for them, where up to 11 students made mistakes in them.

*2) Correct answer rate distribution*

Table VI shows the distribution of the correct answer rates of the VTP instances. 26 VTP instances among 37 achieved 100% correct rate. 10 VTP instances achieved over the 90% correct rate. Thus, they are suitable for self-studies of novice

students. The one instance ID=36 achieved less than 90% correct rate. This instance may be difficult for novice students. In future works, we will improve these instances so that more students can try to solve and answer them correctly.

TABLE VI: CORRECT ANSWER RATE DISTRIBUTION OF INSTANCES

| Average correct answer rate | # of instances |
|---|---|
| 85% – 89% | 1 |
| 90% – 94% | 2 |
| 95% – 99% | 8 |
| 100% | 26 |

*3) Submission time distribution*

Table VII shows the distribution of the numbers of submission times by the instances. 29 instances among 37 are submitted over 50 times and 4 instances are submitted over 75 times. Students submitted answers for one instance two or three times in average so these instances are suitable for the novice students. Four instances are submitted over 100 times so these instances may be difficult for the novice students so we will improve these instances in future works.

TABLE VII: SUBMISSIONS TIMES DISTRIBUTION OF INSTANCES

| Submission time range | # of instances |
|---|---|
| 50 – 74 | 29 |
| 75 – 99 | 4 |
| 100-111 | 1 |
| 112-215 | 3 |

*4) Analysis of hard instances*

Table VIII shows the grammar concepts of the VTP instances where the average correct rate did not reach 100%. They include *data type*, *array*, *string*, *pointer*, and *object-oriented programming specific concepts*, which are generally hard for novice students at studying C++ programming. The teacher should explain them in more comprehensible ways using illustrations or subsidiary tools. Besides, it will be necessary to implement hint functions in the VTP answer interface, to help understanding of these hard concepts, which will be in future works.

TABLE VIII: GRAMMAR CONCEPTS OF HARD INSTANCES

| Instance ID | Topic |
|---|---|
| 2 | data type, size usage |
| 16 | finding average of set of values |
| 17 | two dimensional array |
| 18 | string copy function |
| 20 | combining string and number |
| 21 | pointer |
| 31 | class and object |
| 32 | method in class |
| 34 | encapsulation |
| 35 | polymorphism |
| 36 | inheritance |

## V. CONCLUSION

In this paper, we investigated the effectiveness of the value trace problem (VTP) for self-study of C++ programming through code reading study. We generated 37 VTP instances using simple source codes for basic grammar concepts of C++ programming, and asked 46 students to solve them using the answer interface at home.

By analyzing the answer results of the students, we confirmed the effectiveness of the VTP instances for the novice students. We could detect the understanding levels of them in C++ programming and the hard concepts for them, namely, *data type*, *array*, *string*, *pointer*, and *object-oriented programming specific ones*. In general, they can be difficult for any novice student at studying C++ programming. It is recommended that the teacher should explain them in more comprehensible ways using illustrations or subsidiary tools in the programming course. Besides, a hint function to encourage the understanding should be implemented at the answer interface, which will be in our future works.

In future studies, we will study improvements of VTP instances to hard C++ programming concepts, and implement the hint function in the answer interface. We will also continue generating VTP instances for other grammar concepts or programming topics such as libraries, data structure, and algorithms, and will assign them to students in C++ programming courses.

## CONFLICT OF INTEREST

Authors point out that no conflict of interest.

## AUTHOR CONTRIBUTIONS

Shune Lae Aung conducted the design, implementation and experiments of the proposal and wrote the paper. Nobuo Funabiki gave the idea for this research and supervised whole the activities. Nem Khan Dim and Soe Mya Mya Aye assigned the VTP instances to the students and collected the answers from them. Htoo Htoo Sandi Kyaw assisted writing the paper and analyzing the data.

## ACKNOWLEDGMENT

Authors would like to thank to committee who hold International Conference on Education and Psychological Sciences. We also thank to anonymous reviewers for our paper by making their time. We want to thank to the students who participate to answer VTPs for C++ programming.

## REFERENCES

[1] Top Programming Languages 2021. IEEE Spectrum. [Online]. Available: https://spectrum.ieee.org/top-programming-languages/#toggle-gdpr

[2] N. Funabiki, K. K. Zaw, and N. Ishihara, "Java programming learning assistant system: JPLAS," in *Proc. WSPC*, April 2017.

[3] K. K. Zaw, N. Funabiki, and W.-C. Kao, "A proposal of value trace problem for algorithm code reading in Java programming learning assistant system," *Inform. Eng. Exp*., vol. 1, no. 3, pp. 9-18, Sep. 2015.

[4] N. Funabiki, H. Masaoka, N. Ishihara, I.-W. Lai, and W.-C. Kao, "Offline answering function for fill-in-blank problems in Java programming learning assistant system," in *Proc. ICCE-TW*, pp. 324-325, May 2016.

[5] W.-Y. Hwang, R. Shadiev, C.-Y. Wang, and Z.-H. Huang, "A pilot study of cooperative programming learning behavior and its relationship with students " learning performance," *Comput. Edu*., vol. 58, pp. 1267–1281, 2012.

[6] P. Brusilovsky and S. Sosnovsky, "Individualized exercises for self-assessment of programming knowledge: an evaluation of QuizPACK," *J. Edu. Res. Comput*., vol. 5, no. 6, 2005.

[7] S. S. A. Naser, "Developing an intelligent tutoring system for students learning to program in C++," *Inform. Tech. J.*, vol. 7, no. 7, pp. 1055-1060, 2008.

[8] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppala, "Review of recent systems for automatic assessment of programming assignments," in *Proc. Koli Call. Int. Conf. Comput. Edu. Research*, pp. 86-93, Oct. 2010.

[9] A. K. Jain, M. Singhal, and M. S. Gupta, "Educational tool for understanding algorithm building and learning programming languages," in *Proc. World Cong. Eng. Comput. Sci*., pp. 292-295, Oct. 2010.
[10] N. Ishihara, N. Funabiki, M. Kuribayashi, and W.-C. Kao, "A software architecture for Java programming learning assistant system," *Int. J. Comput. Soft. Eng*., vol. 2, no. 1, Sep. 2017.
[11] Learn C++ Programming. [Online]. Available: https://www.programiz.com/cpp-programming
[12] C++ Language. [Online]. Available: http://www.cplusplus.com/doc/tutorial/
[13] C++ Tutorial – Learn C++ Programming with examples. [Online]. Available: https://beginnersbook.com/2017/08/c-plus-plus-tutorial-for-beginners/
[14] H. Schildt, *C++: A Begineer's Guide*, 2nd edition, McGraw-Hill Education, 2003.



**Shune Lae Aung** received the B.S. degree from Yadanabon University, Mandalay, Myanmar in 2012. She also received the M.Sc degree from Yangon University, Yangon, Myanmar in 2015. She worked as a lecturer in Yangon University from since 2017. Currently, she is a Ph.D candidate in Graduate School of Natural Science and Technology, Okayama University, Japan. Her research interests include educational technology, assistive technology and human computer interaction.

**Nobuo Funabiki** received the B.S. and Ph.D. degrees in mathematical engineering and information physics from the University of Tokyo, Japan, in 1984 and 1993, respectively. He received the M.S. degree in electrical engineering from Case Western Reserve University, USA, in 1991. From 1984 to 1994, he was with Sumitomo Metal Industries, Ltd., Japan. In 1994, he joined the Department of Information and Computer Sciences at Osaka University, Japan, as an assistant professor, and became an associate professor in 1995. In 2001, he moved to the Department of Communication Network Engineering (currently, Department of Electrical and Communication Engineering) at Okayama University as a professor. His research interests include computer networks, optimization algorithms, educational technology, and Web technology. He is a member of IEEE, IEICE, and IPSJ.

**Nem Khan Dim** received the B.Sc and M.Sc degrees in computer science from Yangon University, Yangon, Myanmar, in 2008 and 2011, respectively. She received the Ph.D degree in computer science from Kochi University, Japan, in 2016. She is a lecturer from Department of Computer Studies, Yangon University. Her research interests include assistive technology and human computer interaction.

**Soe Mya Mya Aye** received the B.S. degree in chemistry, Yangon University in 1985. She received M.S. degree in computer science, Yangon University in 2004. She received the Ph.D. degree in computer science, Yangon University in 2012. She is currently a professor from Department of Computer Studies, Yangon University. Her research interests include database and data mining.

**Htoo Htoo Sandi Kyaw** received the B.E. and M.E. degrees in information science and technology from University of Technology (Yatanarpon Cyber City), Myanmar, in 2015 and 2018, respectively. She received Ph.D. from Graduate School of Natural Science and Technology at Okayama University, Japan in 2021. She is currently an assistant professor from Tokyo University of Agriculture and Technology, Japan. Her research interests include educational technology and web application systems.