

IJACT 22-12-19

Design of Learning Process with Code Reconstruction Principle for Non-computer Majors

Hye-Wuk Jung

*Assistant Professor, Dept. of College of Liberal Arts and Interdisciplinary Studies,
Kyonggi University, Korea
wukj@kyonggi.ac.kr*

Abstract

To develop computational thinking skills, university students are learning how to solve problems with algorithms, program commands and grammar, and program writing. Because non-computer majors have difficulty with computer programming-related content, they need a learning method to acquire coding knowledge from the process of understanding, interpreting, changing, and improving source codes by themselves. This study explored clone coding, refactoring coding, and coding methods using reconstruction tools, which are practical and effective learning methods for improving coding skills for students who are accustomed to coding. A coding learning process with the code reconstruction principle was designed to help non-computer majors use it to understand coding technology and develop their problem-solving ability and applied the coding technology learning method used in programmer education.

Keywords: *Coding Education, Non-Computer Majors, Software Education, Code Reconstruction*

1. INTRODUCTION

Currently, various subjects are being taught in universities to enable students to develop computational thinking skills. Among the subjects that non-computer majors must essentially complete, students learn methods for problem solving such as thinking, analysis, and algorithms and computer language[1].

Since programming work using a computer language plays a role in creating solutions for problem solving, students must understand command elements used in programming such as variables, operators, and conditions. The general programming learning method is to learn the functions of individual commands and the grammatical structure to combine and use them. For example, in the basic course of Python programs that many beginners learn, students learn the grammar for basic commands and practice coding methods with programming tasks[2].

Non-computer majors often have little or no programming experience. In particular, in humanities or arts and physical education departments, programming is hardly connected with their majors, so students understand and use the basic contents of programs through liberal arts courses. However, they come to experience difficulties in the process of learning commands and grammar for programming work and applying them to problem solving[3].

Therefore, since computer non-majors need a more efficient way to learn, a method that enables them to acquire programming knowledge and test their ideas in the process of learning what code is and how to apply code to solve problems should be provided.

Manuscript received: October 31, 2022 / revised: November 25, 2022 / accepted: December 3, 2022

Corresponding Author: wukj@kyonggi.ac.kr

Assistant Professor, Dept. of College of Liberal Arts and Interdisciplinary Studies, Kyonggi University, Korea

Copyright©2022 by The International Promotion Agency of Culture Technology. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0>)

The programming learning process includes the following:

- Understanding how computers work and their logic
- Learning the basic commands and grammar of programs
- Understanding the operation of peripheral technologies and problem drawing
- Design for problem solving through programming

This learning process also includes parts where the teacher should provide the solutions necessary for problem solving to beginners who are not familiar with programming. That is, teachers' feedback that enables students to flexibly apply the academic association between their major field and coding to derive problems[4]. However, since one-to-one learning guidance in the classroom has a time limit, students' self-directed learning ability to solve problems should be developed[5]. In addition, it is difficult to perfectly master all the broad programming-related skills with classes. Therefore, a learning method that will enable students to acquire coding knowledge from the process of understanding, interpreting, changing, and improving the code by themselves is necessary.

From this point of view, it is necessary to apply the coding learning method used by developers or people majoring in the learning process for non-computer majors. The clone coding, refactoring coding, and coding methods using reconfiguration tools for programmer education can enable students to learn related skills and improve software quality by understanding and changing the implemented software structure. However, non-computer majors may feel that these methods are difficult because they need to have basic knowledge of programs or be able to adapt to the use of tools.

Therefore, this study proposes a coding learning process for non-computer majors with the code reconstruction principle which is designed for novice programming learners to develop their problem-solving abilities.

2. CODING LEARNING METHODS FOR PROGRAMMER EDUCATION

2.1 Clone Coding

Clone coding is a method of programming the structure of an existing system and is used to learn services such as YouTube or Instagram while implementing those services[6]. In this method, students can increase their understanding of program development through the process of creating frequently used services, and can develop their application ability by changing the technology that is actually applied.

The advantage of clone coding is that it helps improve coding skills because learners learn the grammar of the necessary programming language and develop software centered on core service technologies while making identical websites or app services that are actually being.

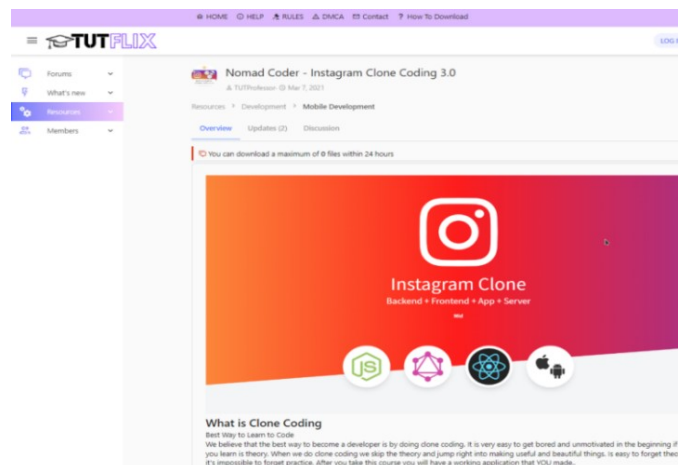


Figure 1. Tutflix site that provides cloning coding learning [8]

Clone coding can be learned from a domestic nomad coder website, and can be helped by Tutflix, an online learning network, through various libraries and video education contents[7]. Figure 1 shows the Tutflix site that provides cloning coding learning for Instagram[8]. In the clone coding lectures, students can learn how to use tools, coding patterns, and libraries necessary for development, as well as the process of reorganizing code or efficiently modularizing it, and the overall content of software development.

2.2 Software Refactoring

Refactoring is a method of improving the structure of software and finding and fixing vulnerabilities through code reorganization while maintaining the original functions, and is used when updating existing software or adding new functions[9]. Among the refactoring techniques proposed by Martin Fowler, Figure 2 shows the Move Method, which is the process of creating a new method in the class used the most frequently, and then replacing the old method into a reference or removing it entirely[10].

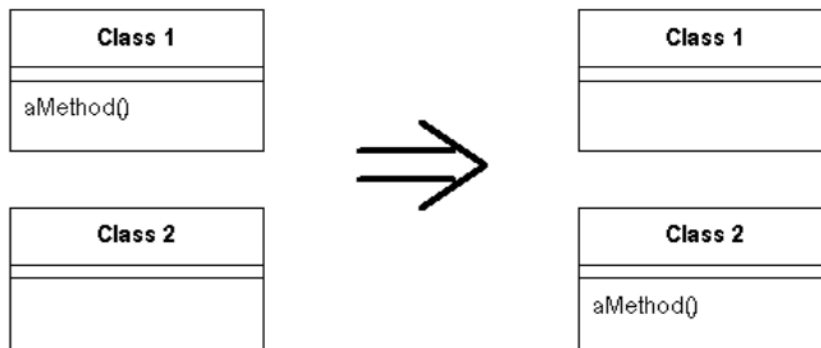


Figure 2. Process of Move Method [10]

This method is used by software developers to improve the quality of code because it has the advantage of simplifying the code to increase readability and reduce complexity. In addition, Son et al. applied refactoring to the education method so that students can easily understand and design programs to control hardware such as robots[11].

2.3 Code Completion Tool

The coding method using the code completion tool is a method of acquiring programming knowledge while completing and modifying part of the program source code. Garner proposed a code writing solution that can reduce the cognitive load felt by computer programming beginners in the process of understanding and writing source code[13]. As shown in Figure 3, CORT(Code Restructuring Tool), a visual programming interface, was designed and a restructuring tool was developed that enables students to add, modify, and learn code[14]. They can efficiently learn coding by using CORT to complete partially missing codes and to ask and answer questions about programming problems.

Harms developed a curriculum using code puzzle completion problems for beginners in programming to provide a tool for them to learn coding methods by themselves[15]. The code puzzle was created by applying a block-based programming approach, thereby simplifying the syntax of the source code, and using a storytelling context. Students can acquire coding knowledge through the process of finding statements to solve requirements and reusing example-based codes.

Coding education for programmers consists of processes to implement, modify, or review systems actually serviced, or restructuring the presented program. These methods are practical for students familiar with coding and can be effectively applied to improving coding skills. However, since the students should have basic knowledge of the program or be able to adapt to using the tool, beginners may take a lot of time and

feel difficulty. Therefore, this study is intended to design a learning process that enables students to become familiar with coding work through a method of acquiring basic knowledge of programming languages and restructuring codes according to difficulty.

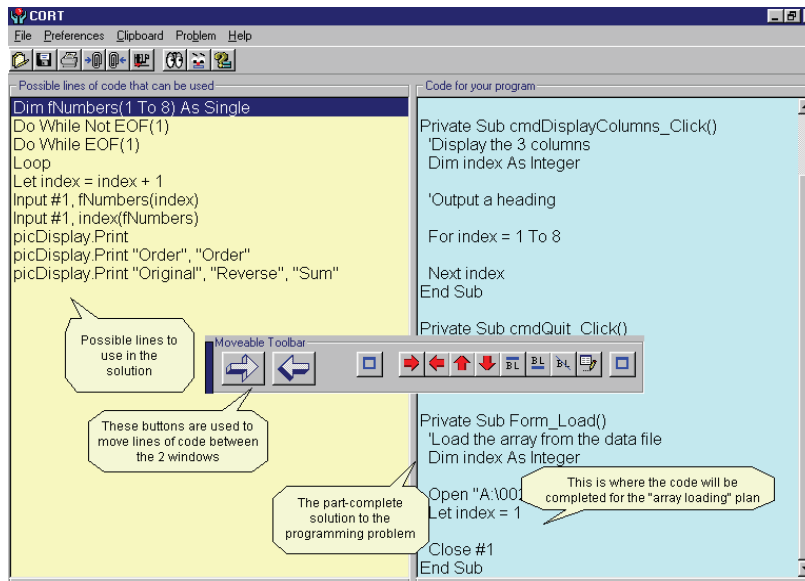


Figure 3. Code Restructuring Tool (CORT) [14]

3. DESIGN OF CODING LEARNING PROCESS BY APPLYING CODE RECONSTRUCTION PRINCIPLE

The process of the coding learning with the code reconstruction principle applied proposed in this study is as shown in Figure 4.

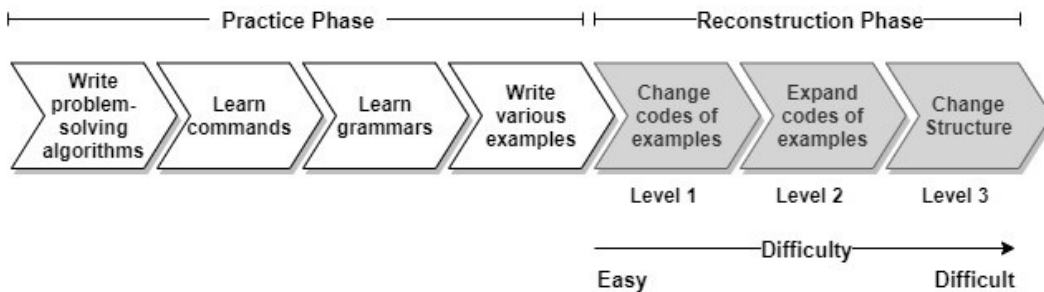


Figure 4. The process of the coding learning with the code reconstruction principle

In the practice phase, students learn how to write algorithms for problem solving, learn commands and grammars of programming languages, write various examples, and acquire basic knowledge related to coding. In the reconstruction phase, which is composed of levels 1 to 3 according to the level of difficulty, the code of the example is changed and expanded, or the entire structure of the code is changed.

Code reconstruction is performed by classifying the processes in levels 1 to 3 in the contents of the learning area including input/output, variable, operator, list, conditional statements, iteration statements, functions, and algorithms as shown in Figure 5.

Each phase of learning is carried out during the period from week 1 to week 15 with the contents shown in Table 1. The teacher adjusts the period of each reconstruction level according to the learner's level of understanding and gives conditions for code reconstruction.

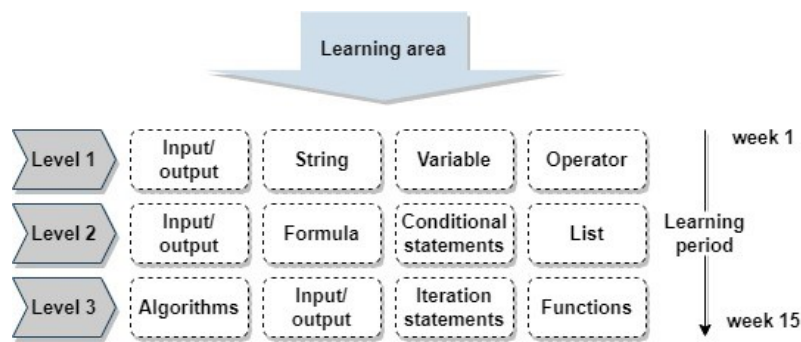


Figure 5. Code reconstruction levels by the learning area

Table 1. The learning contents by the reconstruction levels

Level	Learning Contents
1	Change strings used in input/output functions → Change variable names used in input functions and formulas → Change operators in formulas
2	Add input/output functions → Add formulas using operators → Add conditional statements □ Add conditional statement using list
3	Write problem-solving algorithms → Change input/output structure → Construct iteration statements → Change the code structure using functions

In Level 1, students learn the principles of input and output programming, change strings and variable names used for input and output, and practice basic code writing. Also, they change the operator in the formula and check the execution result values.

In Level 2, students write the given example, add input/output functions, formulas, and conditional statements, and expand and change the conditional statements using list.

Level 3 is the process of generally changing examples including the contents of all the learning areas, and students plan the contents of the program and write the algorithm for problem solving. They change the overall structure of the code by using the input/output structure, iteration statements, and functions.

The learning process with the code reconstruction principle is designed so that novices in programming can experience code reconstruction step by step by modifying and reviewing examples by difficulty according to the contents of the beginning, middle, and last part of the coding course for one semester. In addition, the students can carry out coding work more flexibly than the method using a standardized tool by reviewing the coding techniques learned during the semester through the reconstruction principle, applying the algorithms planned and changing the code structures by themselves. The process can be used to learn and understand various programming techniques and develop the ability to solve problems by observing example codes and reorganizing them according to the learner's thoughts.

4. CONCLUSION

This study proposed a coding learning process with the principle of code reconstruction that was designed to help non-computer majors develop their problem-solving abilities and the process applied clone coding, refactoring coding, and coding methods using the reconfiguration tools used in programmer education. In the proposed method, beginners in programming can modify and review examples by difficulty according to the learning content for a semester. And they experience code reorganization work by step, apply the algorithm which was planned by them, and change the code structure. Therefore, this process can be used to understand and change the software structure, learn programming techniques, and develop problem-solving ability. Studies on the analysis of applied cases of actual coding education classes conducted through the learning process designed in this study and the review of their effectiveness are needed in the future.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2015R1D1A1A01061064 and NRF-2020S1A5A8042850).

REFERENCES

- [1] H. W. Jung, "Exploring the Convergence Possibility of Learner's Major Field in Software Project - Focus on the Non-Computer Major Freshmen -," *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 6, No. 2, pp. 263-270, 2020. DOI: <https://doi.org/10.17703/JCCT.2020.6.2.263>
- [2] H. W. Jung, "A Study on Coding Education for Non-Computer Majors Using Programming Error List," *International Journal of Advanced Culture Technology*, Vol. 9, No. 1, pp. 203-209, 2021. DOI: <https://doi.org/10.17703/IJACT.2021.9.1.203>
- [3] H. W. Jung, "A Case Study of Python Programming Error in an Online Learning Environment," *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 7, No. 3, pp. 247-253, 2021. DOI: <https://doi.org/10.17703/JCCT.2021.7.3.247>
- [4] Lombardi, M., "Authentic learning for the 21st century: An overview," *Educause Learning Initiative(ELI)*, 2007.
- [5] H. W. Jung, "A study on basic software education applying a step-by-step blinded programming practice," *Journal of Digital Convergence*, Vol. 17, No. 3, pp. 25-33, 2019. DOI: <https://doi.org/10.14400/JDC.2019.17.3.025>
- [6] Nomad Coders. <https://nomadcoders.co>
- [7] Tutflix - Free Education Community. <https://tutflix.org>
- [8] Nomad Coders - Instagram Clone Coding 3.0. <https://tutflix.org/resources/nomad-coder-instagram-clone-coding-3-0.3340>
- [9] Tutflix - Free Education Community. <https://tutflix.org>
- [10] Matheus, P., Anderson, U., Ana, C. B., Daniel, O., Alessandro, G., Jens, K., and Emilio, A., "Behind the Intents: An In-depth Empirical Study on Software Refactoring in Modern Code Review," in *Proc. 17th International Conference on Mining Software Repositories(MSR '20)*, Vol. 17, 2020. DOI: <https://doi.org/10.1145/3379597.3387475>
- [11] Martin Fowler, *Refactoring: Improving the Design of Existing Code (2nd Edition)*, Addison-Wesley Professional Press, 2018.
- [12] H. S. Son, W. Y. Kim, H. Y. Ahn, and R. Y. Kim, "Applying Design Pattern & Refactoring on Implementing RTOS for the Small Educational Multi-Joint Robot," *The Journal of The Institute of Webcasting, Internet Television and Telecommunication*, Vol. 9, No. 3, pp. 217-224, 2009.
- [13] Stuart Garner, "An Exploration of How a Technology-Facilitated Part-Complete Solution Method Supports the Learning of Computer Programming," *Issues in Informing Science and Information Technology*, Vol. 4, pp. 491-501, 2007. DOI: <https://doi.org/10.28945/966>
- [14] Stuart Garner, "A Quantitative Study of a Software Tool that Supports a Part-Complete Solution Method on Learning Outcomes," *Journal of Information Technology Education: Research*, Vol. 8, pp. 285-310, 2009. DOI: <https://doi.org/10.28945/698>
- [15] Harms, K. J., Code Puzzle Completion Problems in Support of Learning Programming Independently, McKelvey School of Engineering Theses & Dissertations. 240, 2017. DOI: <https://doi.org/10.7936/K74B2ZQP>