International Journal of Membrane Science and Technology, 2023, Vol. 10, No. 1, pp. 122-127

A Study on Improvement of SW Structure Understanding through Gyeonggi Dream University

Hanyong Choi

Department of Software Convergence Shinhan University, Korea.
E-mail: hychoi@shinhan.ac.kr

Abstracts: Software education is included in the regular curriculum for a certain period and is thoroughly conducted following the software education operation guidelines. Gyeonggi-do Province operates "Gyeonggi Dream University" for high school students belonging to each local office of education. In this study, we compare and analyze the understanding of software between students who have completed software training courses at "Gyeonggi Dream University" and those who have not, and those who have currently majored in software at the university. After comparing the scores obtained from the pre and post test, the analysis was confirmed to be statistically meaningful through the T-test. As a result, the index value of those who took classes through public education among students in the group with an improved understanding of software was 0.9. However, it was confirmed that the improvement level of the Gyeonggi Dream University students was 2.3, and the index value was 4.8. Therefore, it was confirmed that the 'understanding software through coding' course at 'Gyeonggi Dream University' has a positive effect on understanding the basic structure of software.

Keywords: Coding, Software Understanding, T-test, Software Structure, Gyeonggi Dream University.

1. INTRODUCTION

As the software convergence industry develops, the 4th industrial revolution is rapidly approaching. The convergence software industry is driving the Fourth Industrial Revolution by applying software to non-software domains to create added value[1]. As most developed countries, including the United States, scramble to participate, the demand for jobs related to computers and software is also increasing[2,3]. Therefore, since the software industry is in the spotlight as an important growth engine for the country's and society's development, the school curriculum is also changing rapidly for demand in the industry. Until now, software-related education has focused on training developers, focusing on the department of Computer Science & Engineering at universities[4-6]. However, it is now changing to develop basic capabilities such as learning the basic principles of computers through algorithms and understanding software utilization by coding themselves. Since 2014, software-related education has begun in the UK, and European countries are gradually taking the same steps[7-9]. In South Korea, the Ministry of Education announced an amendment to the integrated curriculum (2015 revised curriculum) in 2015, noting the importance of software[10]. According to this, software-related education has been included in the regular curriculum since 2015. In order to expand to elementary, middle, and high schools from 2018, research and leading schools for software education were selected and operated on a trial basis from 2015, and various educational models were developed for field application. Regular courses at schools selected for research and guidance for software education included more than a certain amount of time in software education, which was thoroughly conducted following the Software Education Operational Guidelines. In addition, Gyeonggi-do Province operates Gyeonggi Dream University for high school students. That is a program in which high school students in Gyeonggi-do take courses offered by universities or institutions that have signed a business agreement with the Gyeonggi-do Office of Education, providing opportunities to think about their careers through experiences. In this study, the subjects are students who have completed software training courses at "Gyeonggi Dream University," those who have not, and those who have currently majored in software at the university. Check the understanding of software among high school students who have taken the "Understanding the Software World through Coding" course over the past three years (6 semesters) and want to pursue a career in software-related fields[11, 12]. Then the study would like to analyze and compare the changes in the understanding of software of first-year students currently majoring in computer engineering at university and present directions for future software education.

2. COMPOSITION OF SW EDUCATION METHOD

2.1. Software Education Method in Korea

In Korea, software-related education is being conducted as follows. Elementary schools focus on finding ways to use software to solve problems while relating software to real life. To this end, it starts with the concept of an algorithm and then gradually expands to understanding software. In middle school, students learn basic elements necessary to understand software such as computers, algorithms, programs, and information ethics, and practice coding to set and solve simple problems on their own. In high school, the level is even higher, focusing on learning the structuring, abstracting, modeling, and evaluation process of information to design more sophisticated algorithms while developing software to solve various problems and developing information's concept and convergence ability. Software education is divided into three categories: First, students learn software, information ethics, and information processing concepts in the life and software category. Then, in the algorithms and programs category, students learn information, algorithms, and concepts and structures of software. Lastly, in the computing and problem-solving category, students learn the principles and information of processing, algorithms, program design, and the activities of converging the obtained results based on computing thinking. The elements of software education learned in the public education process in South Korea can be found in Table 1.

Table 1. Contents of software education.

Category	Elementary school	Middle school	High school
Life and software	Me and the software	Utilization and Importance of Software	Information Acquisition Life and Computing
	Information ethics	information and ethics	Information ethics
		Configuration of information devices and exchange of information	Operation and Processing of Information Devices
Algorithms and programs	Experience and Resolution of Problems	Types and Structure of Information	Presentation and Management of Information
		Understand Computing	What is Computational Thinking?
	Experience the Algorithms	Understand Algorithms	What is the Algorithms?
	Experience the Programming	Understand Programming	Understand Programming
			Programming for Troubleshooting
Computing and problem-solving		Computational Thinking for Troubleshooting	Computational Thinking and Convergence Activities

2.2 Software Education Method in Gyeonggi Dream University

"Gyeonggi Dream University" aims to provide opportunities to develop imagination and creativity by providing education tailored to students' careers and aptitudes, away from textbook-oriented learning methods. It is possible to realize student-centered education that students can independently choose and experience while studying convergence topics through their own experiences. Education methods are divided into visit type, regional type, and online type. In the visiting type, students visit universities or institutions in person to learn. Regional types are studied at designated facilities in the region, not at the institution to which the instructor belongs. Moreover, online types can be taken on an online platform so that students can learn anytime, anywhere. It runs for eight weeks on a semester basis. Classes are divided into weekdays and weekends and are operated for high school students regardless of grade. Gyeonggi Dream University uses software that is easily accessible to people around them as an educational tool, allowing them to learn how computers work and software concepts while practicing thinking like computers. For education, students could understand the software through the kiosk system that they often meet at restaurants, shopping centers, hotels, and airports.

3. ANALYSIS METHODS AND GROUP COMPOSITION

3.1 Learning Contents

3.1.1 Overview of the 'Understanding the Software through Coding' course

The contents of the 'Understanding Software through Coding' process conducted by 'Gyeonggi Dream University' are as follows. The focus was on gaining basic coding experience before full-scale SW development by selecting a language with excellent accessibility and affinity. Most of the participating students were familiar with using the Internet, so they learned the kiosk software system using HTML and Javascript using the MS Visual Studio Code development tool. This course aims to understand coding while experiencing direct programming for software development. Software technology is being used in various places, such as computers, robots, airplanes, and drones, as well as the Internet and mobile. Therefore, this course aims to understand the principles of software production and operation through coding practice. To this end, students will practice creating the internal logic of kiosks installed in vending machines and fast food restaurants, and the characteristics of the course are as follows. High school students should be able to understand the configuration and development environment of software, and to this end, they learn the basic process of software development through programming theory and practice [13-14]. Understand the coding method of software that runs in various environments, such as the Internet and mobile, and carry out a mini-project that directly creates the code of the core logic necessary for kiosk production. The career guidance process conducted after completing the course was introduced as follows. Students can enter convergence-related departments such as artificial intelligence, big data, and IoT, as well as computer engineering, which is the core department of software, which is the center of the Fourth Industrial Revolution and work for software companies such as game companies, Google, Samsung Electronics, and Naver after graduation.

3.1.2 Learning Contents by Session

The "Understanding Software through Coding" course was divided into 17 sessions for 8 weeks, and the contents of each lesson are as follows. (see Table.2) Students' types, characteristics, and understanding of software through orientation were evaluated in advance. Next, while investigating the satisfaction of students who completed eight weeks of education, it was quantitatively and qualitatively evaluated whether their understanding of the software improved.

Table 2. Learning Content by Week.

Sessions	Learning Contents	Session Method
1	Course introduction and orientation	Course Orientation, Discussion on Understanding Software
2	How to install and use software development tools and write basic code	How to install and configure code for software development tools
3	To create a basic structure for a programming language	How to configure basic code structures
4	Understanding how data is represented and computing operations	Data representation and processing methods for kiosk development
5	Optionally process and create an action statement	How to handle menus in kiosks and beverage vending machines
6	Multiple processing of multiple selection menus	How to handle different menus in kiosks
7	Repeated code execution	How to code repetitive order processing on an unattended system
8	Complex structural control and mini-projects	Create a simple kiosk system for mini project Discussion among participants on the project

3.2 Research Analysis Method

In this study, first-year students attending S University in Gyeonggi-do were selected as subjects to be compared to those who took 'Understanding Software through Coding.' In addition, in order to increase reliability, a preliminary fact-finding survey, research topics, and research tasks were selected. In addition, the study was conducted after establishing an operation plan based on this. In order to confirm the research results, a survey was conducted on the software-related curriculum conducted in the public education process, and the contents learned at Gyeonggi Dream University. In addition, the performance of SW education was verified by setting up an experimental group and a control group. Test papers and questionnaires were prepared to check students' understanding of the software, and scores were compared before and after, then statistically meaningful through the T-test. Finally, to analyze learners, the difference in scores between "groups with an improved understanding of software" and "groups with no improved understanding of software" was compared. In order to confirm the understanding status of software education, a test paper consisting of a total of 36 questions based on a Likert scale of 5 points was prepared. The answer to the question is selected from ① to ⑤. The larger the number, the closer it is to the positive. Through the questions, the student's thoughts on software understanding education were asked, and the understanding according to the contents of this course was evaluated simultaneously. Compare the pre and post-scores of the experimental and control groups; then, it was used as data to determine whether it was statistically meaningful through the T-test.

4. RESEARCH RESULTS AND ANALYSIS

In Figure 1, to analyze learners, (a) it was divided into "groups with an improved understanding of software" and (b) "groups with no improved understanding of software," and the results were compared. (a) There is a significant difference in understanding between students who did not take the course through Gyeonggi Dream University(GD0) and course-completed students (GD1).

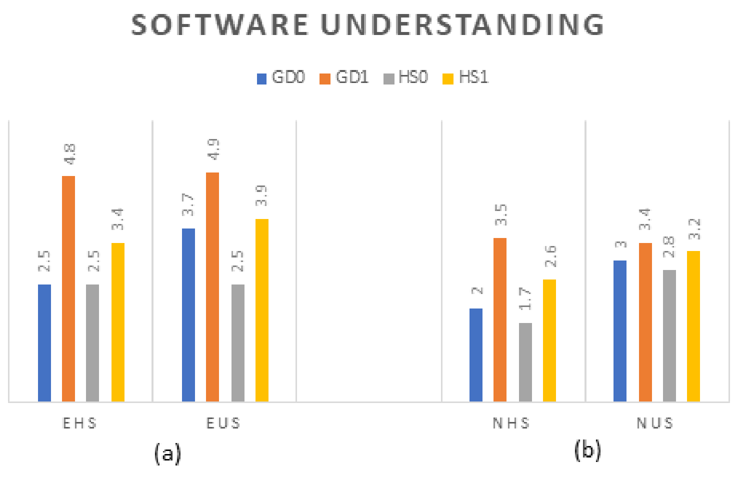

Figure 1. Comparison of Software Understanding.

However, it can be seen that there is a clear difference between students who have not taken software-related education in formal education(HS0) and students who have experienced it (HS1). Among the "groups with an improved understanding of software" students, the index value of the public-educated students was 0.9, while the improvement of the Gyeonggi Dream University students was 2.3. In addition, the index value is 4.8, which can be confirmed as a high result. That reflects that Dream University is more effective, although it is helpful to increase understanding of software through public education. Even in the 'groups with no provided understanding of software, the results of Dream University students were relatively high compared to those who did not. However, students in "groups with no provided understanding of software" may be unwilling to learn about the software, so additional group characteristics analysis is needed.

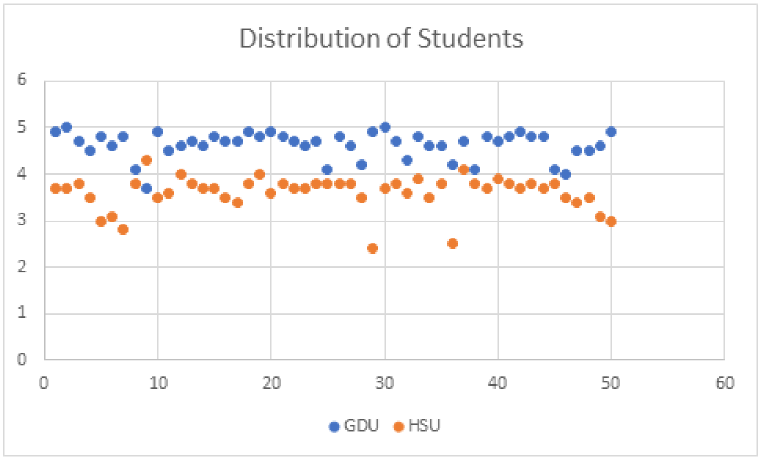

Figure 2. Distribution of Students.

Figure 2. shows the distribution of values that measure the understanding between university students who studied software in "Dream University" (GDU) and students completing only public education before went to university. It was found that the understanding of the software was bisected between the two groups, as students in the upper group already fully understood the software before choosing a major, while students in the lower group began to understand the software after entering university. According to these results, additional research is needed on the major's suitability and the relationship with the career. In addition, unlike the computational thinking discussed earlier, it can be expected that understanding software has a positive effect on understanding major skills.

5. CONCLUSION

This study analyzed the effect of the 'understanding software through coding' process operated by Gyeonggi Dream University on software understanding. It was confirmed that information-related parts of SW education greatly influenced most students. It was also confirmed that information-related SW education improves students' understanding of the software. It was also confirmed that analyzing the SW information-related process could be affected by computational thinking and attitudes depending on the class type, class time, and level of knowledge that students previously had. Among the "groups with an improved understanding of software" students, the index value of the public-educated students was 0.9, while the improvement of the Gyeonggi Dream University students was 2.3. In addition, the index value is 4.8, which can be confirmed as a high result. That shows that software education in most schools also helps students understand software, but Dream School is effective. Moreover, the study shows the distribution of values that measure the understanding between university students who studied software in "Dream University" (GDU) and students completing only public education before went to university.

The 'understanding software through coding' process operated by Gyeonggi Dream University was analyzed to positively affect the understanding of the basic structure of software. Therefore, it is considered important to devote the maximum time allowed in the regular curriculum and provide SW-related education tailored to the eye level of elementary, middle, and high schools to provide an environment for continuous learning. In addition, various learning and activity methods should be developed in areas related to SW education, and securing a certain amount of time in the regular curriculum should be sufficiently reviewed.

6. ACKNOWLEDGEMENTS

This work was supported by the Shihan University Research Fund, 2023

REFERENCES

[1] Min Xu, Jeanne M. David&Suk Hi Kim, “The Fourth Industrial Revolution: Opportunities and Challenges”, International Journal of Financial Research, Vol.9, No. 2, pp.90-95, 2018

[2] D. Bar, J. Harrison and L. Conery, “Computational Thinking: A Digital Age Skill for Everyone,” Learning & Learning with Technology, Vol.

38, NO. 6, pp. 20-23, 2011.

[3] Hmelo-Silver, Cindy E., “Problem-Based Learning: What and How Do Students Learn?,” Educational Psychology Review, 16 (3): 235, 2004.

[4] B. B. Levin, “Energizing teacher education and professional development with problem-based learning,” Alexandria, VA: Association for Supervision and Curriculum Development, 2001.

[5] Maika I. Patino, Peggy Kraus, Martin A. Bishop “Implementation of patient education software in an anticoagulation clinic to decrease visit times for new patient appointments” Patient Education and Counseling Volume 102, Issue 5, pp 961-967, 2019

[6] F. J. García-Peñalvo, “What computational thinking is,” J. Inf. Technol. Res., vol. 9, no. 3, pp. 5–8, 2016.

[7] F. J. García-Peñalvo and A. J. Mendes, “Exploring the computational thinking effects in pre-University education,” Comput. Hum. Behav., vol. 80, pp. 407–411, Mar. 2018.

[8] P. Martín-Ramos et al., “First exposure to Arduino through peercoaching: Impact on students’ attitudes towards programming,” Comput. Hum. Behav., vol. 76, pp. 51–58, Nov. 2017.

[9] F. J. Garcí-Peñalvo, D. Reimann, M. Tuul, A. Rees, and I. Jormanainen, An Overview of the Most Relevant Literature on Coding and Computational Thinking With Emphasis on the Relevant Issues for Teachers. Brussels, Belgium: TACCLE3 Consortium, 2016.

[10] Kim, Kapsu, “A Recognition Analysis of Elementary Teachers for Software Education of 2015 Revised Korea Curriculum”, Journal of The Korean Association of Information Education, Vol.20, Issue 1, pp.47-56, 2016.

[11] Yangsoon Kim, “The Problem/Project-Based Learning (PBL/PjBL) at Online Classes”, International Journal of Advanced Culture Technology, Vol.9 No.1, pp.162-167, 2021.

[12] Suendarti, M. ‘’Protecting Our Planet: The Vital Role of Carbon Sequestration in Combating Threats to Environmental Sustainability’’. Pakistan Journal of Life & Social Sciences, Vol. 21 No. 1, 2023.

[13] Ibrahim, H., Indonesia, M., & Zain, M. M. ‘’Embracing Change: Analyzing the Impact of Farmers’ Knowledge and Risk Perception on the Adoption of Innovations in Agribusiness’’. International Journal of Applied and Physical Sciences, Vol 8, pp. 22-29, 2022.

[14] Wang, C. H., & Wu, K. C. ‘’Interdisciplinary Learning of Low-Code Development Platform Programming with Dual Coding Theory-A Case Study of Agilepoint NX’’. Journal of ICT, Design, Engineering and Technological Science, pp. 21-25, 2022.

DOI: https://doi.org/10.15379/ijmst.v10i1.1437