applied sciences

Article

Developing a Learning Data Collection Platform for Learning Analytics in Online Education

JuGeon Pak [1], JooHwa Lee [2] and MyungSuk Lee [1,*]

1 Tabula Rasa College, Keimyung University, Daegu 42601, Korea; corea@kmu.ac.kr
2 Department of Computer Engineering, Keimyung University, Daegu 42601, Korea; jhlee71@kmu.ac.kr
* Correspondence: mslee@kmu.ac.kr

Abstract: During the COVID-19 pandemic, most education has been conducted through online classes. While feedback and interaction between students and instructors are significant in programming education or engineering practice, online education today cannot satisfy these aspects of learning. Therefore, this study proposes a learning support system for programming education and presents the results of designing and implementing this system. The proposed system consists of an online development environment module, a learning monitoring module, and a learning support module. It also provides a web-based programming environment, real-time chat and code mirroring, error guide messages and related lectures, e-learning quizzes, and learning activity analysis features. The system standardizes the development environment between the instructor and students, helps students take the initiative in solving errors, and enables code-oriented interactions between the instructor and students. It also collects data from all learning situations in the database. Conducting a big data analysis with the collected data will enable individual guidance for students by finding errors that frequently occur in programming and recommending learning materials to solve them.

Keywords: online education; learning analytics; programming education; data collection; platform; learning data

check for updates

Citation: Pak, J.; Lee, J.; Lee, M. Developing a Learning Data Collection Platform for Learning Analytics in Online Education. *Appl. Sci.* **2022**, *12*, 5412. https://doi.org/10.3390/app12115412

Academic Editor: Valentino Santucci

Received: 30 April 2022
Accepted: 24 May 2022
Published: 26 May 2022

1. Introduction

As Industry 4.0 creates new added value by converging information and communication technologies, such as artificial intelligence, the Internet of Things, cloud computing, and big data, we are rapidly changing into a software-oriented society, in which software is the center of value creation. At a time when the significance of software education is increasing every day, the government and educational institutions are continuing their efforts to nurture talent and introduce various programming education courses [1].

At the same time, the COVID-19 pandemic has brought many changes in our lives and society. Due to the prolonged pandemic and the government's intensive social distancing measures, we are entering the era of the contact-free society, and the field of education is no exception to this trend. As such, there is a growing interest in non-face-to-face online education that enables learning to take place anytime, anywhere, without constraints on time and space, unlike traditional classroom-oriented learning methods [2].

Although non-face-to-face online education is considered future-oriented education due to advantages such as improving the flexibility and convenience of learning and providing opportunities for self-directed learning, it also limits the interaction between instructors and students [3]. According to the findings of an investigation of complaints made by students in actual online classes, many students (21.1%) have pointed to the lack of interaction with instructors [4]. Various methods have been attempted to improve interaction, including using video chat services such as ZOOM, real-time Q and A through chatting, and inducing student participation through assignments or quizzes, but these have not produced satisfying results.

These problems may become more serious in practical fields of education, such as programming. Due to the nature of programming education, practice is significant, and code-oriented real-time interaction is essential. However, the conventional methods used in non-face-to-face online education have the following limitations. First, one of the difficulties students face in programming is finding and solving errors that occur in the programming process [5]. They need feedback from the instructor to resolve these errors, but there are limitations to communication through traditional methods, such as chatting. Second, active participation is required to learn programming syntax and understand the procedure, such as students entering and writing the code themselves, even if code examples are provided [6,7]. However, conventional methods that use assignments or quizzes can only determine whether students submitted their code, not whether they actively participated or not.

Therefore, this study proposes a learning support system for non-face-to-face online programming education and presents the results of designing and implementing this system. The proposed system consists of an online development environment module, a learning monitoring module, and a learning support module. The online development environment module provides an online development environment based on Docker containers, where students can practice Python, JavaScript, and C languages. The learning monitoring module collects and monitors the learning situation, such as the code input status, error status, and questions, and the learning support module transmits learning contents, quizzes, and error message descriptions to students that have been pre-registered by the instructors.

The advantages of applying the proposed system to non-face-to-face online programming classes are as follows. First, programs often depend on platforms or operating systems, but the proposed system can unify the development environment between instructors and learners by providing an online development environment. This is very useful in non-face-to-face online education, where students participate in classes using their personal computers. Second, the instructor can easily assess active student participation by regularly monitoring their code input status and progress. Third, providing additional explanations and references for programming errors in real time helps students take the initiative in solving them. Fourth, code mirroring enables code-oriented interactions between the instructor and students.

2. Related Works

As learning activities in physical schools and classrooms move to online environments, various learning support services have been introduced through trial and error in terms of technical and content preparation. Accordingly, many attempts have been made to encourage students to participate and improve their achievement in software education. Programming experience is significant in software education, so many studies have been conducted on increasing code-oriented interaction.

Quizzes are one of the best methods of enhancing students' programming experience and evaluating their achievement [8,9]. A wide variety of research has been performed that focuses on this aspect, including studies that proposed and developed an online automatic evaluation system with question banks for main topics in data structure which evaluates the answers submitted by students in terms of memory usage, execution time [10], and research on the design and implementation of automated assignment assessment systems using basic commands such as Commit and Pull provided by Git (a version control tool) to register and submit assignments and GitHub repositories to store them [11–13]. GitHub is a useful tool for programming education. It can be used as a code editor, as it offers syntax highlighting features, and is also helpful for team assignments, as it is optimized for collaboration. However, it also has many limitations; for one, it is impossible to compile and execute codes and examine the progress of syntax units, because we can only monitor when the students submitted their assignments.

A study by Kim et al. (2020) proposed a code-related conversation service for online programming education to increase interaction around code [14]. This service is a code-based conference support system in which the source code by a specific student is posted in an online space, where both the instructor and students participate, and participants conduct code reviews and Q and As. It also provides features to allow the insertion and sharing of notes on specific codes for effective code reviews. A web-based integrated development environment (IDE) was proposed to provide an IDE to students to standardize the development environment between the instructor and students, offer learning content, and collect learning statistics.

CODECHUM [15] offers a web-based IDE for C programming classes and provides leaderboards based on student performance history to encourage participation. Class-Code [16] is a system proposed for JavaScript classes. Students can take online tutorials prepared by instructors, write and submit example programs, and refer to other students' solutions. Other studies in this area have used commercial online IDE, such as REPL.IT, CODECHEF, and IDEONE for online classes [17–19].

3. Learning Analytics System

3.1. Functional Requirements

Functional requirements were analyzed to design and develop an efficient learning analytics system for non-face-to-face online programming classes. The requirements and design results are as follows.

First, the system needs to provide students with the same development environment as the instructor and an environment that is not dependent on students using a specific platform or operating system. This is because syntax may differ depending on the version of the interpreter (compiler) used, and additional modules or libraries that can be installed may also vary depending on the operating system. This problem can be solved through a web-based IDE.

Second, one of the challenges experienced by students is finding and solving the errors that occur while executing source codes. Error messages generated by interpreters (compilers) in traditional IDE are difficult to understand for beginners who are just starting to learn to program. If an error occurs in a face-to-face class, it can be solved immediately by showing the error message with the source code to the instructor, but it takes more time to resolve errors in online classes due to the limits on communication. Therefore, the system needs to provide error messages generated by the interpreter (compiler) together with error guidance messages and reference tutorials to help students understand and solve the errors in real time. The system also needs a chat function that enables real-time communication between the instructor and students and a real-time student source code mirroring function.

Third, one of the ways to improve student learning and achievement in programming classes is to encourage their participation by motivating them to enter and execute the code by themselves, even if there is an example source code disclosed by the instructor, so that they can become familiar with the programming syntax and structure. Therefore, the system needs to recognize whether students are actively participating in real time. However, participation in online classes can be indirectly determined by information, such as the students' log-in and source submission statuses, but it is impossible to assess how actively they are participating in the class in real time. As such, there is a need to monitor the students' code input progress in real time.

3.2. System Structure

The proposed system is composed of an online development environment module, a learning monitoring module, and a learning support module.

3.2.1. Online Development Environment Module

The online development environment module provides a web-based online development environment to support programming classes. The online development environment module in this study was developed for Python, JavaScript, and C language education. It collects information about student learning activities during classes, including student login information, code input status and code content, the occurrence of errors and their content, and questions. The learning activity information collected in real time is transmitted to the learning monitoring module.

Another function of the online development environment module is to immediately provide learning content and feedback to students according to the learning guidance rules already received from the learning support module. The student interface of the online development environment module is shown in Figure 1.

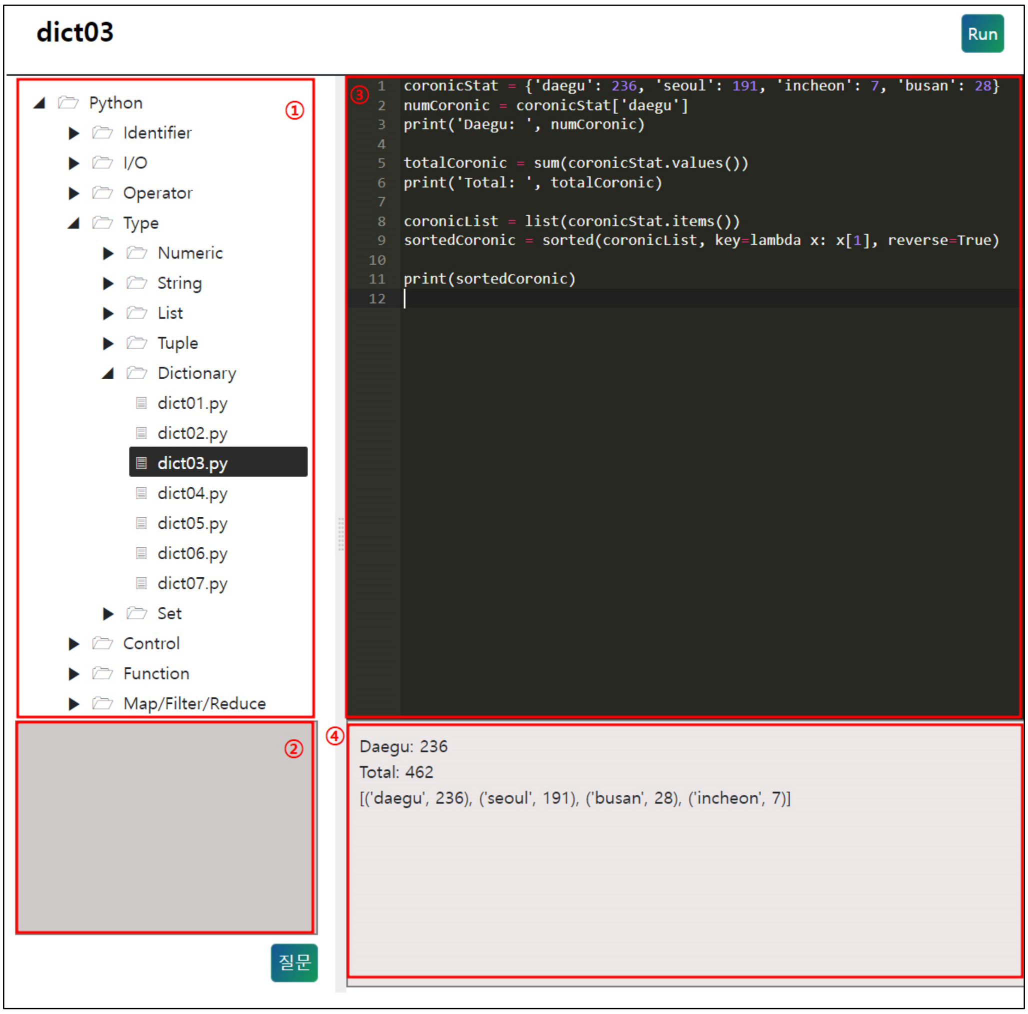

Figure 1. Student interface. ① The file window shows the example source code for each unit or chapter uploaded in advance by the instructor. It is a window for selecting and editing or executing a specific file. ② The question window is for entering and sending questions to the instructor. ③ The editor window is for editing the source code, and provides syntax highlighting features. Any code changed by student input is transmitted to the instructor through the learning monitoring server. ④ The results window outputs the results of executing the source code.

The online development environment module consists of a client that provides the development environment interface to the students and a server that returns the results of compiling student source codes and transmits and receives data to and from the learning monitoring module. The online development environment module was implemented with

NodeJS v12.22.11 based on the Ubuntu 18.04 LTS Linux operating system and uses REST API and Socket.IO v4.4.0 to communicate with the learning monitoring module.

3.2.2. Learning Monitoring Module

The learning monitoring module stores the learning activity information received from the online development environment module in the database, analyzes real-time and long-term accumulated learning activity information, visualizes the analysis results and provides them to the instructor. An example of the real-time learning activity analysis results is shown in Figure 2.

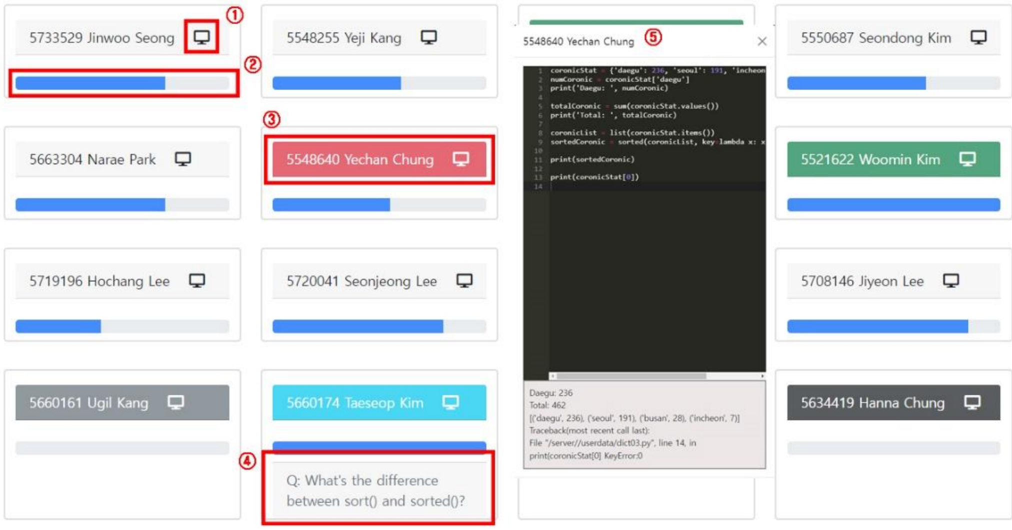

Figure 2. Real-time learning activity analysis window. ① The monitoring button allows instructors to remotely monitor students' source code input status from their PC. ② The progress shows the progress of each student compared to the target source code in the form of a bar chart. ③ The progress status is visualized in different colors by assessing the progress/waiting/completed/error/question/absence status according to the code input status regularly received from the students. ④ The question content shows the students' questions. ⑤ The student modal window monitors the student's source code and shows it to the instructor.

Figure 3 shows the results of analyzing student learning status based on the data accumulated in the database, including the number of assignments submitted, number of posts, number of questions, total logged-in time, activity time, and attendance. The learning monitoring module is based on the Ubuntu 18.04 LTS Linux operating system, implemented with NodeJS, and uses MySQL 8.0.27 as the database.

3.2.3. Learning Support Module

The learning support module provides a web-based interface to support the instructor's programming class, through which the instructor can register learning contents such as course contents, error guidance messages, and quizzes. The interface of the learning support module is show Figure 4.

The error guide messages registered in Figure 4 are transmitted to the online development environment module and shown to the students in a modal form. Figure 5 shows the occurrence of a 'KeyError'. The system displays a pop-up window with the error guide message related to a 'KeyError' registered in advance by the instructor and a link to the lecture or tutorial in the online development environment of the student.

Python C Lang

■ STUDENTS

ID	NAME	Dept.	Status	Submission	Post	Inquery	Total Residence Tim	Total Activity Time	Attendance
5733529	Jinwoo Seong	International Trade	ACT	5/5	3	0	850/825	375/825	11/11
5548255	Yeji Kang	Accounting	ACT	5/5	0	1	873/825	383/825	11/11
5719134	Sangho Lee	Business Administrati	ACT	4/5	1	1	1011/825	297/825	11/11
5550687	Seondong Kim	Public Administration	INACT	5/5	0	3	847/825	284/825	10/11
5663304	Narae Park	Public Administration	ACT	5/5	3	0	863/825	327/825	11/11
5548640	Yechan Chung	Game Mobile Engine	INACT	5/5	30	2	881/825	251/825	11/11
5485551	Taesan Ahn	Industrial Design	ACT	5/5	0	0	904/825	287/825	11/11
5521622	Woomin Kim	Public Administration	ERR	5/5	1	0	933/825	315/825	11/11
5719196	Hochang Lee	Business Administrati	ACT	5/5	0	1	847/825	311/825	11/11
5720041	Seonjeong Lee	Computer Engineerin	ABS	5/5	0	6	866/825	296/825	11/11
5719239	Homin Ahn	Computer Engineerin	INACT	5/5	0	02	794/825	197/825	10/11
5708146	Jiyeon Lee	Computer Engineerin	ACT	4/5	0	0	871/825	225/825	11/11
5660161	Ugil Kang	Business Administrati	INACT	5/5	2	0	853/825	271/825	11/11
5660174	Taeseop Kim	Business Administrati	INACT	5/5	1	5	848/825	309/825	11/11
5660922	Yuri Lee	International Trade	INACT	3/5	1	0	871/825	315/825	11/11
5634419	Hanna Chung	Business Administrati	ACT	5/5	0	0	840/825	322/825	11/11

Figure 3. Results of analyzing student learning status.

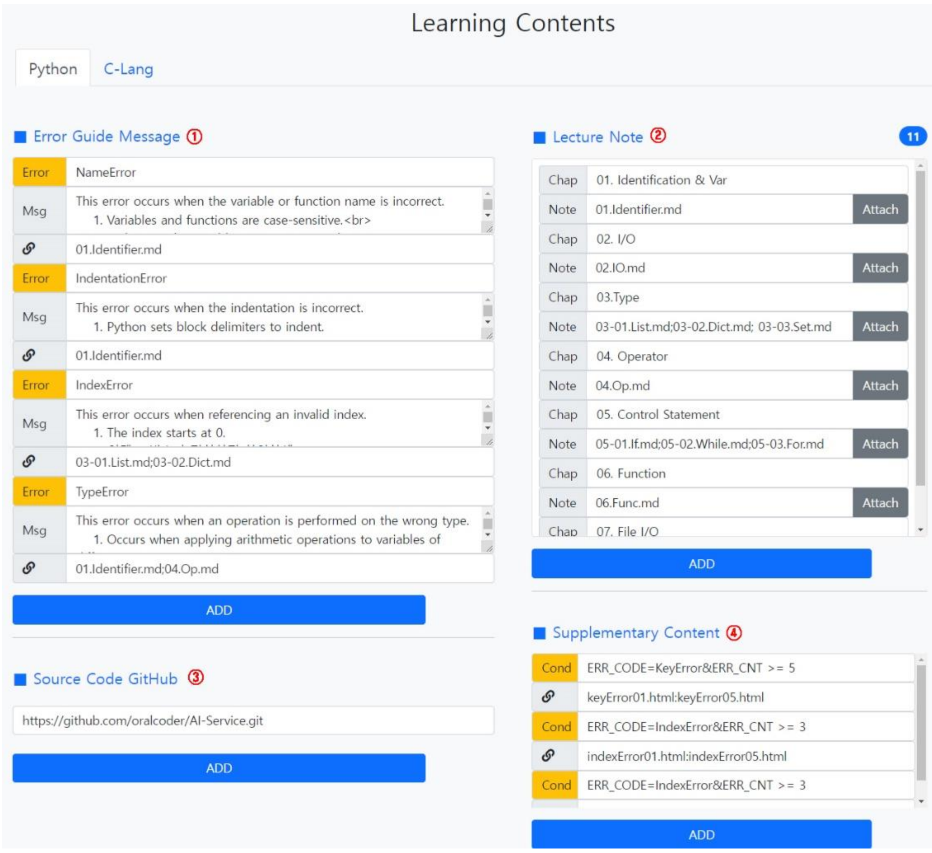

Figure 4. Learning support module interface. ① The error guide message allows the instructor to register guide messages for the students for each type of programming error and reference tutorials. ② The lecture syllabus is for the syllabus. ③ The source code repository is the source code repository address where the example sources for the lecture are registered. ④ Additional learning materials are where the instructor enters error occurrence conditions and materials that require additional learning, such as student error occurrence type and frequency.

If the learning monitoring module determines that additional learning is necessary (such as when there are repeated similar errors), students receive supplementary learning content (such as quizzes) registered in the learning support module, as shown in Figure 6.

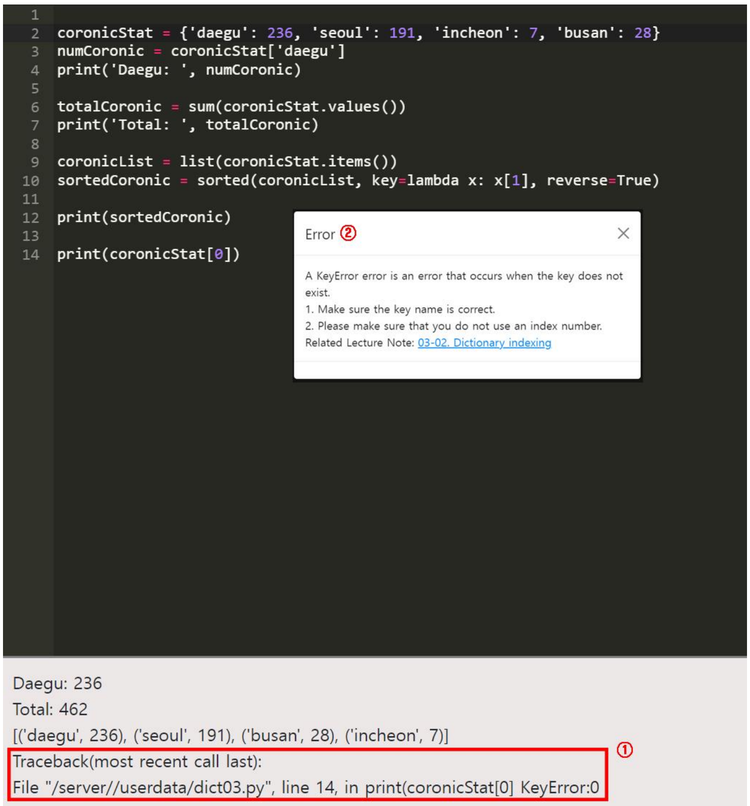

Figure 5. Related error guide message and links to lecture and tutorial. ① The error message returned by the interpreter (compiler) when a source code error occurs. ② Error guidance messages for each error pre-registered by the instructor in the learning analytics cloud server.

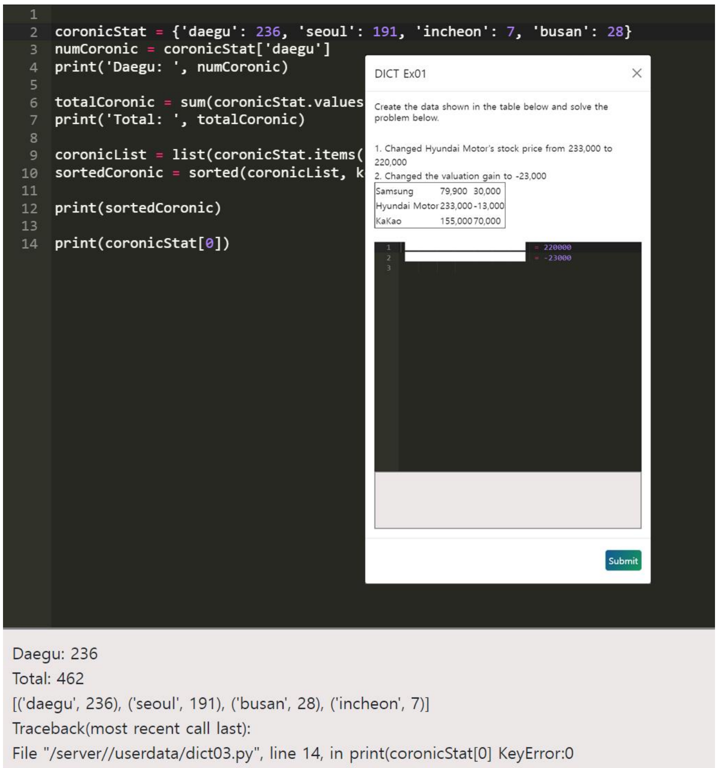

Figure 6. Quiz–additional learning content.

The learning analytics module is based on the Ubuntu 18.04 LTS Linux operating system, implemented with NodeJS, and uses Socket.IO for real-time communication with the online development environment module.

4. Materials and Methods

4.1. Research Questions

Our goal in this study is to improve the effectiveness of learning in programming courses. We conducted an experiment to evaluate the following research questions.

1. Does the proposed system help students to take the initiative in solving programming errors?
2. Is the system beneficial for online programming courses?

4.2. Course and Participants

We designed and conducted an experiment during the first semester of 2022 to evaluate whether the proposed system is beneficial for learning programming. The experiment involved 40 first-year undergraduate students attending the Introductory C Programming course (20616-06).

We divided the students into two groups: the experimental group (n = 20) and the control group (n = 20). In the first phase (week 2–4) and third phase (week 9–11) of the term, both of the groups did not receive any additional learning materials. In the second phase (week 5–7), the experimental group received additional learning materials, such as error guide messages and supplementary learning contents, by the system, while the control group did not. The eighth week was excluded in the experimental due to the midterm week.

4.3. Procedure

4.3.1. Provision of Lecture Materials

The teacher registered error guide messages and supplementary learning contents for each error type and uploaded quizzes for each chapter to the learning support module. The module provides a web-based interface and is only accessible by teachers. The registered and uploaded lecture materials are stored in a database.

4.3.2. Practice Exercises

The participated students had three or four programming quizzes in every lecture except the first week. They wrote source codes to answer the quizzes and submit them using the online development environment module. In this procedure, the error history and submission state were monitored and stored in a database by the learning monitoring module.

The students can verify whether the result of the code execution is successful or not through the module. If an error is generated, the experimental group will receive additional learning materials, such as error guide messages and supplementary learning contents by the system, while the control group receives an original error messages generated by the interpreter (compiler).

4.4. Data Collection

In order to compare the number of errors in the experimental group and control group, the learning monitoring module kept error data in a database every time when the students executed their source code. We extracted the stored data from the database using the learning monitoring module.

5. Experimental Result

During the experiment, the participated students had three or four programming quizzes in every lecture, except the first week. We gathered the error status data for each group via the learning monitoring module.

We evaluated the average number of errors during each programming quiz. The result of this evaluation is shown in Figure 7.

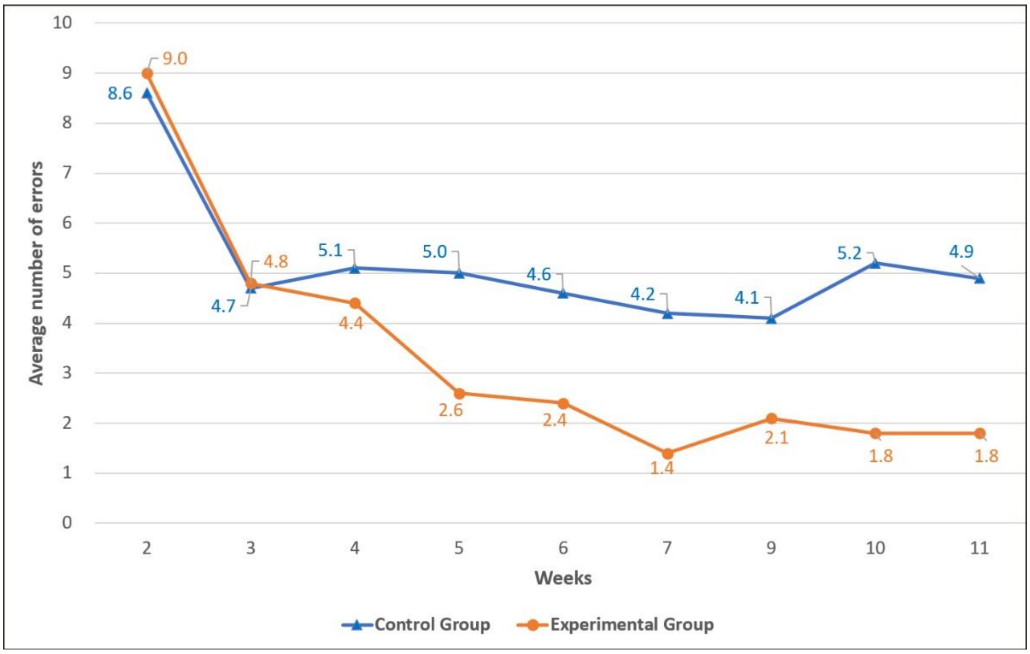

Figure 7. Comparison between two groups in average number of errors.

According to Figure 7, both of the groups generated a large amount of errors, and the difference between two groups was slight in the first phase. In the second phase, however, the experimental group, which received additional learning materials by the system, had much fewer errors than the control group, and this trend continued into the last phase. We think that this is because the experimental group was well trained in the second phase. The results indicate that the proposed system is useful in helping learners reduce programming errors and improving the academic achievement of programming majors.

6. Conclusions and Future Work

In this paper, we propose a learning support system for non-face-to-face online programming classes and present the results of designing and implementing this system. The proposed system consists of an online development environment module, a learning monitoring module, and a learning support module. It provides a web-based programming environment, real-time chat and code mirroring, error guide messages and related lectures, e-learning quizzes, and learning activity analytics features. The system standardizes the development environment between the instructor and students, helps students take the initiative in solving errors, and enables code-oriented interactions between the instructor and students.

To verify the proposed system, we applied the system to a programming course. We divided the students into two groups, where one is supported by the proposed system and the other is not, and compared between the two groups the average number of errors during programming quizzes. The result indicate that the proposed system is beneficial for the programming course.

There are still some limitations of this study. Firstly, it was a small-scale and short-term study. More students should be involved, and long-term experiments are necessary in the further study. Secondly, the proposed system only provides additional learning content according to simple and standardized rules pre-registered by the instructor, so it has limitations when it comes to offering customized content based on each student's learning pattern and participation. Currently, the proposed system is being applied and used in university programming lectures. Based on the learning activity information accumulated by the system, future work will include research on AI-based learning achievement prediction models and providing customized learning content for each student.

Author Contributions: Conceptualization, J.P. and M.L.; methodology, J.L.; software, J.P.; validation, J.P., J.L. and M.L.; writing—original draft preparation, J.P.; writing—review and editing, M.L.; visualization, J.L.; supervision, M.L.; project administration, J.P.; funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Koran government (MSIT) (No. 2020R1F1A1070800).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, A.H. Domestic research trends analysis of software education. *J. Korean Assoc. Educ. Inf. Media* **2018**, *24*, 277–301. [CrossRef]
2. Paudel, P. Online education: Benefits, challenges and strategies during and after COVID-19 in higher education. *Int. J. Stud. Educ.* **2021**, *3*, 70–85. [CrossRef]
3. Mukhtar, K.; Javed, K.; Arooj, M.; Sethi, A. Advantages, Limitations and Recommendations for online learning during COVID-19 pandemic era. *Pak. J. Med. Sci.* **2020**, *36*, 27–31. [CrossRef] [PubMed]
4. Lee, Y.S.; Shin, D.K. An investigation of the implementation of online classes in the untact era caused by the COVID-19 pandemic. *J. Curric. Eval.* **2020**, *23*, 39–57. [CrossRef]
5. Safei, S.; Shibghatullah, A.S.; Mohd Aboobaider, B. A perspective of automated programming error feedback approaches in problem solving exercises. *J. Theor. Appl. Inf. Technol.* **2014**, *70*, 121–129.
6. Höök, L.J.; Eckerdal, A. On the bimodality in an introductory programming course: An analysis of student performance factors. In Proceedings of the 2015 International Conference on Learning and Teaching in Computing and Engineering, Taipei, Taiwan, 9 April 2015.
7. Freeman, S.; Eddy, S.L.; McDonough, M.; Smith, M.K.; Okoroafor, N.; Jordt, H.; Wenderoth, M.P. Active learning increases student performance in science, engineering, and mathematics. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 8410–8415. [CrossRef] [PubMed]
8. Denny, P.; Dahlstrom, D.; Purchase, H.C.; Luxton-Reilly, A.; Hamer, J. Self-predicted and actual performance in an introductory programming course. In Proceedings of the 15th Annual Conference on Innovation and Technology in Computer Science Education, Ankara, Turkey, 26 June 2010.
9. Harvey, L.; Aggarwal, A. Exploring the Effect of Quiz and Homework Submission Times on Students' Performance in an Introductory Programming Course in a Flipped Classroom Environment. In Proceedings of the 2021 ASEE Virtual Annual Conference, Virtual, 17 May 2022.
10. Kim, S.S.; Oh, S.H.; Jeong, S.S. Development and application of problem bank of problem solving programming using online judge system in data structure education. *J. Korean Assoc. Comput. Educ.* **2018**, *21*, 11–20.
11. Kim, C.Y.; Oh, J.H.; Jeong, H.J.; Hong Ha, M.J.; Hong, S. Design and Implementation of Automated Programming Assignment Assessment Systems for GitHub Repositories, Koran Institute of Information Scientists and Engineers. In Proceedings of the 2019 Korean Software Conference Proceeding, Seongnam-si, Korea, 18–20 December 2019; pp. 760–762.
12. Barra, E.; López-Pernas, S.; Alonso, Á.; Sánchez-Rada, J.F.; Gordillo, A.; Quemada, J. Automated Assessment in Programming Courses: A Case Study during the COVID-19 Era. *Sustainability* **2020**, *12*, 7451. [CrossRef]
13. Feliciano, J.; Storey, M.A.; Zagalsky, A. Student Experiences Using GitHub in Software Engineering Courses: A Case Study. In Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering Companion, Austin, TX, USA, 14 May 2016.
14. Kim, J.E.; Jeon, Y.W.; Hong, S. Design and Implementation of Code-related Conversation Service for Online Programming Education. In Proceedings of the 2020 Korean Computer Conference (KCC), Busan, Korea, 2–4 July 2020; Volume 47, pp. 892–894.
15. Maranga, J.; Matugas, L.; Lim, J.; Lyn, S.R.C. Codechum: An online learning and monitoring platform for C programming. In Proceedings of the 6th International Conference on Educational Technologies 2019, Hong Kong, China, 8–10 February 2019; pp. 109–118.
16. Ryo, S.; Kato, J.; Yatani, K. ClassCode: An interactive teaching and learning environment for programming education in classrooms. *arXiv* **2020**, arXiv:2001.08194.
17. Rahman, M.M.; Paudel, R.; Sharker, M.H. Effects of Infusing Interactive and Collaborative Learning to Teach an Introductory Programming Course. In Proceedings of the 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, 16 October 2019.
18. Zubrycki, I.; Granosik, G. Teaching robotics with cloud tools. In Proceedings of the International Conference on Robotics and Education, Sofia, Bulgaria, 26 April 2017.
19. Kurniawati, A.; Kusumaningsih, A.; Sophan, M.K. Visualization code tools for teaching and learning introductory programming. In Proceedings of the 2018 2nd International Conference on Informatics for Development, Yogyakarta, Indonesia, 7 November 2018.