International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 3
ISSN: 2088-5334

Design and Implementation of a Programming Automatic Assessment System in Jupyter Notebook

HakNeung Go [a], Seong-Won Kim [b], Youngjun Lee [a,*]

[a] *Korea National University of Education, Cheongju, 28173, Republic of Korea*
[b] *Silla University, Sasang-Gu, Busan, 46958, Republic of Korea*
Corresponding author: [*] yjlee@knue.ac.kr

***Abstract*—Learning programming is challenging. So, computer educators have developed various tools to help students. In this paper, we have developed a tool that combines the advantages of a Programming Automatic Assessment (PAA) system and Jupyter Notebook (JN) to support learning programming. The design direction of this system is free to use, easy to set up, and supports interactive computing. The Programming Automatic Assessment in Jupyter Notebook (PAAinJN) is available free of charge using the assessment module released on Git and the personal JN. The initialization is completed by executing in a code cell with two lines of code that downloads and executes the assessment module. In an interactive computing environment, presenting problems, writing code to be evaluated, and evaluating code can be executed in the code cells, and the problems and the results of the assessment are presented in the code cell outputs. The performance was verified by the examples presented in a high school informatics textbook using the programming automatic assessment system as teaching learning material. In addition, we propose a way to develop teaching-learning materials using PAAinJN in consideration of teachers and students and a way of distributing and collecting teaching-learning materials using the free Learning Management System. PAAinJN is expected to help students learn programming by eliminating assessment and feedback delays through PAA while learning to program in an interactive computing environment.**

***Keywords*—Automatic assessment system; Jupyter notebook; interactive computing; Python.**

Manuscript received 10 Dec. 2022; revised 13 Mar. 2023; accepted 27 Apr. 2023. Date of publication 30 Jun. 2023.

I. INTRODUCTION

With the advent of The Fourth Industrial Revolution, the importance of computing skills and software has garnered much attention around the globe. In particular, computing assists us in solving efficiently and effectively the difficult things that humanity faces. Therefore, education is being developed to set computational thinking as a core competency in domestic and foreign education and to cultivate computational thinking [1], [2].

Computational thinking is the ability to build a problem-solving model by collecting, extracting, and analyzing the data necessary to address complex and diverse problems, and to implement these models in a language that computers can understand [3], [4]. Programming education that recognizes and solves a given problem through programming has been studied to affect not only computing thinking skills but also higher cognitive development among the various studies on developing computing thinking skills [5], [6], [7].

Programming requires programming knowledge and the skill to utilize programming tools. Learning programming is challenging for many learners, and computing educators have developed various tools to help them learn about programming [8], [9]. Learners address programming tasks and learn programming through feedback on these tasks. However, manual grading of students' assignments by instructors causes delays between assignment submission and evaluation, and if there are many tasks to be evaluated, instructors have difficulty providing feedback to learners [10]. As a tool to support this, a Programming Automatic Assessment (PAA) system has been developed, and learning can be supported by automatically assessing student-submitted code and providing immediate feedback [11], [12]. However, for learners, presentation-based material has no interaction, and the programming tools are separated, resulting in poor participation in learning. The Jupyter Notebook (JN) is a tool that addresses this, which consists of cells that can include text, video, code, description, and formulas, allowing learners to read explanations, practice

code, and check the results. Using this, teachers can provide students with materials for self-directed learning, which can improve their participation and motivation [13], [14], [15], [16].

Tools have been developed to enable automatic assessment in the JN that combines the advantages of the JN with PAA systems to provide instant feedback and interactive computing. However, issues such as cost, setup difficulties, and interactive computing restrictions have occurred [17], [18]. Therefore, this study analyzes the PAA system, programming education conducted in the JN, and the automatic assessment system in the JN, and then designs and implements a system for Programming Automatic Assessment in the Jupyter Notebook (PAAinJN).

II. Materials and Method

A. Programming Automatic Assessment System

The PAA is a method of evaluating the output value obtained by executing the code after entering the input value into the code submitted by the learner, compared to the test case defined by the teacher [8]. It is suitable for students to develop programming skills by addressing various and difficult assignments in subjects such as programming methodology, data structure, and algorithms.

The PAA was developed to compensate for the following disadvantages of manual assessment. The first is the occurrence of a delay between assignment submission and feedback. The second is much time is required to grade the students' assignments. The third is an emphasis on the standard solving method. The fourth is a focus on style rather than correctness. Finally, the subjective nature of the intervention of the evaluator [11].

The PAA comprises problems, scoring data sets, automatic assessment, and user service environments. Problems should be structured, formal, and clear-cut in input and output and create commensurate scoring data. Scoring data sets consist of input data, answer data pairs, and evaluate the accuracy of the code through the longitudinal configuration of input data and answer data. Each scoring data set evaluates the algorithm's efficiency with a lateral configuration that increases the range of input data. The automatic assessment program compiles the code submitted by the student, inputs the generated executable files in order, checks the match of the value output from the execution files and answer data, and determines the correct and incorrect answers. The user service environment is a context for problem presentation, writing code and submitting, and feedback, and is mainly implemented on websites [19], [20].

In Korea, informatics teachers have operated PAA through a website and use it for informatics classes and informatics competitions. In particular, informatics science subjects in the national curriculum [21] present a PAA as a teaching and learning method. There are several studies on cognitive and affective areas using PAA, and it said that programming achievement [22] and immersion [23] were improved compared with the comparative group, and the number of attempts of the automatic assessment system was confirmed to affect computational and logical thinking, as well as immersion and self-efficacy.

B. Programming Education Using the Jupyter Notebook

The JN integrates text cells, code cells, and code cell outputs into a single document and can execute code cells and instantly output results, including formatted text and visual graphs. It provides an interactive and repetitive computing environment similar to the Read Evaluation Print Loop (REPL). In addition, the results can be reproduced through the cells and source code of the JN [13], [14]. The interactive computing environment of the JN and the composition of learning materials that can reproduce results are used in programming education using the JN. Teachers provide learning materials for the learner, including explanations and mathematical formulas, Python codes, and graphs on the JN. Learners can check the results by executing code in a code cell, and because the code and description are integrated into one document, they can easily practice what they have learned in class [24], [25]. Programming education using the JN is generally provided in the form of lecture notes, submitting tasks or projects, and flipped learning materials [24].

C. An Automatic Assessment System for Evaluation on Jupyter Notebook

The JN was used to handle teaching and learning materials to present explanations and write code in a programming class. However, there were difficulties in assessing code or distributing and collecting teaching and learning materials. To address this problem, various tools have been developed to assess codes automatically and distribute and collect teaching and learning material from the JN. Nbgrader uses Jupyter Hub as a server, making it easy to distribute and collect the JN, and automatically assesses the code written In the JN [26]. However, the server must be operated, and resource problems may arise depending on the number of learners and server size. A setup is also required to create a config file for server construction. Software as a Service (SaaS), such as Codle, CoCalc, and Vocareum, can utilize the cloud-based JN and PAA provided by the company. However, there is a cost to use it. Otter and OK can be used for free, but they cannot automatically assess themselves and must utilize other modules through the config. UNCode provides PAA through web pages like SaaS, but it can only be used if it belongs to it [13], [17], [27].

In terms of being free to use, easy to set up, and providing interactive computing, the comparison is as follows in Table I.

TABLE I
COMPARISONS FROM THE PERSPECTIVE OF BEING COST-FREE, EASY TO SET UP, AND PROVIDING INTERACTIVE COMPUTING

	Cost-free	**Easy to set up**	**Interactive computing**
nbgrader	△*	X	○
SaaS	X	○	○
Otter·OK	○	X	X
UNCode	△**	X	○

* Server deployment costs and resource troubleshooting costs
** Free but available only if belong to affiliates

D. Design

By combining the advantages of the JN and the PAA systems presented in previous studies, the PAAinJN was designed. The design direction was free of charge, the setup was simple, and interactive computing was enabled.

To operate the PAAinJN for free, instead of operating SaaS or Jupyter Hub, the assessment module was developed so that programming could be automatically assessed in a personal JN and released to Git for anyone to use. In addition, a Learning Management System (LMS), which can be used for free, was used to distribute and collect teaching and learning materials.

The setup of PAAinJN is to download the assessment module to a personal JN and operate it in the kernel instead of writing a config file. To simplify this process, the learner can complete the setup by executing the setup code on the JN code cell. Problem presentation, evaluation code writing, and outputting assessment results are not performed outside the JN but can be performed in the code cell and the code cell output of the JN using the interactive computing characteristics of the JN.

III. Results and Discussion

A. Composition

The PAAinJN was developed by Python 3.8 and consisted of a problem module (problem.py) and an evaluation module (code_check.py). The problem module (problem.py) consists of a function defined to output the problem to the code cell output, a variable storing the problem (question_00), a variable storing the scoring data set (test_set_00), a file name to be evaluated, a problem variable of the file to be evaluated, and a set of scoring data files to be evaluated as a dictionary.

The assessment module consisted of importing the problem module, the library necessary for assessment, and the functions necessary for assessment. The functions required for assessment include code pre-processing (code_arrange())), code conversion (code_convert())), execution and error verification (error_check()) and checking the answer (code_test()) functions, which were integrated into code_check() functions. Table II describes the PAAinJN module, and the functions and variables included in the module.

TABLE II
ASSESSMENT MODULE CONFIGURATION

File	Function ·variable	Description
problem.py	Question ()	Presenting a problem
	question	Problem
	test_set	Scoring data set
	meta_data	Including scoring data, problem, and assessment file name
code_check.py	code_arrange()	Code pre-processing
	code_convert()	Code conversion
	error_check()	Execute the code and error validation
	code_test()	Check the answer
	code_check()	Consolidate the assessment process into one function

B. Representation and Application

The PAAinJN implemented the setup, problem presentation, writing assessment code, and outputting assessment results in the code cell, as shown in Fig. 1. Fig. 2 shows the process of problem presentation, writing assessment code, and outputting assessment results in web browsers composed of code cell and code cell output and kernels of the JN.

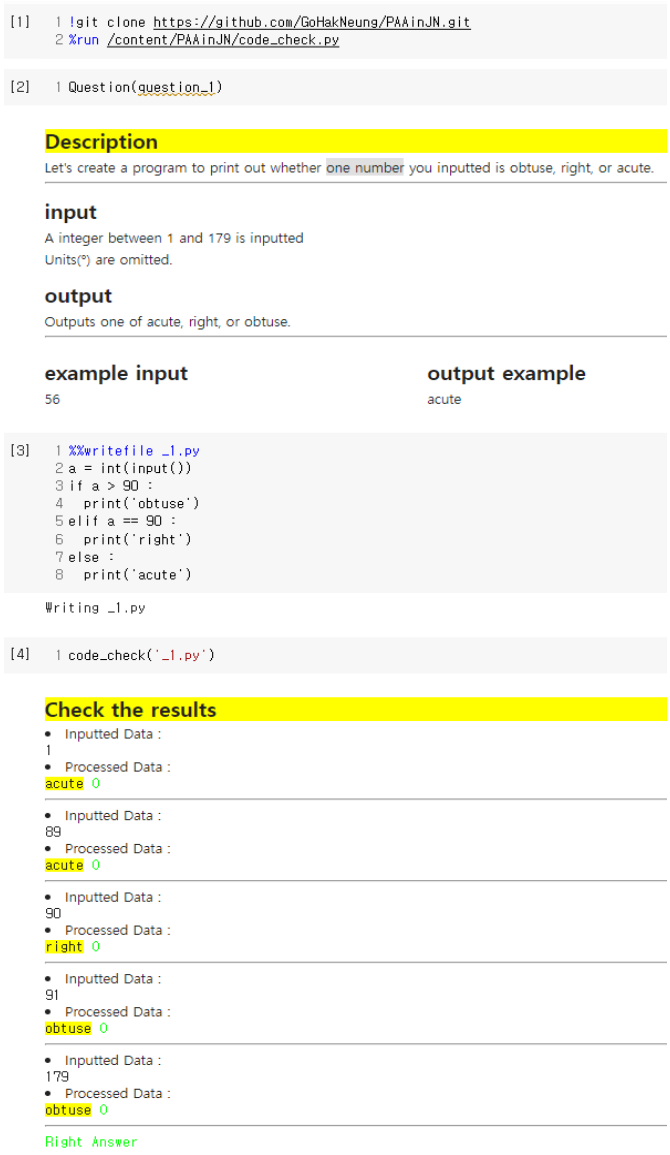

Fig. 1 PAAinJN implementation

1) Setup: The setup is to download the assessment module released on git using the shell command and execute it using the magic command of the JN to operate in the kernel [28]. The code is shown in Fig.3. The first line is the code for downloading the assessment code, the second line is the code for executing the assessment module, and the file path is based on Colab, a JN provided by Google.

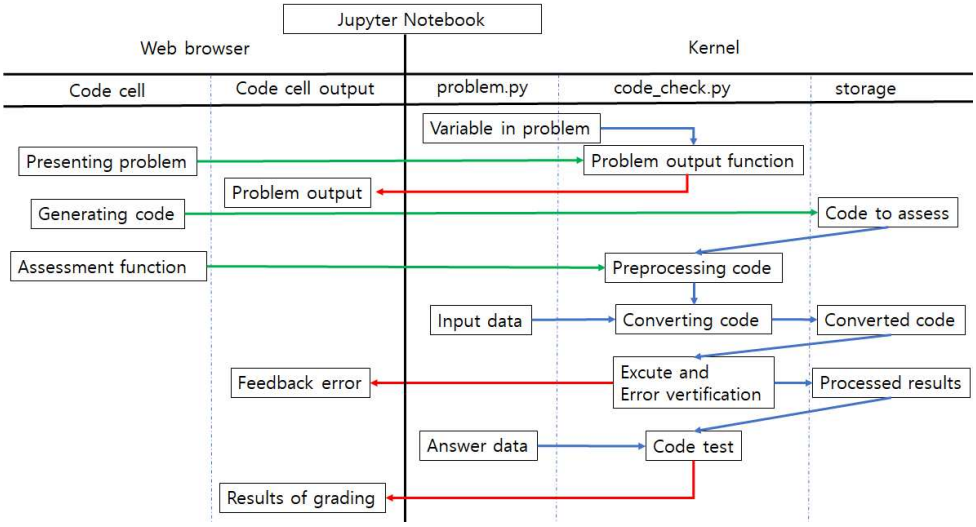

Fig. 2 PAAinJN Operating principles

```
1 !git clone https://github.com/GoHakNeung/PAAinJN.git
2 %run /content/PAAinJN/code_check.py
```

Fig. 3 Setup code in the Jupyter Notebook

2) Problem Presentation: For problem presentation, the problem stored in the variable was output to the code cell output using the problem output function. To structure and show the background description, an explanation, and an example of an input and output included in the problem. The problem was saved as below along with the HTML tag and was saved as shown in Fig. 4. To render and output HTML tags; it was defined as shown in Fig. 5 using the display and HTML functions of the iPython.display module [29].

```
question_0 ='''<h2 style = "background-color:yellow; ">Description</h2>
<p>Let's create a program to print out whether
<span style = "background-color : #E0E0E0">one number</span>
you inputted is obtuse, right, or acute.</p>
<HR>
<h2> input
</h2>
```

Fig. 4 Problems with HTML tags

```
1 from IPython.display import display, HTML
2 def Question(question_) :
3   display(HTML(question_)
```

Fig. 5 The function that outputs a problem to HTML formatting.

3) Assessment code writing: Writing assessment code was used by the magic command (%writefile filename) that inserts the code cell into the file. After writing the assessment code and executing the code cell, it was saved into the kernel, which was used for code assessment. Fig. 6 shows an example of writing and storing the code to be evaluated using the magic command.

```
1 %%writefile _0.py
2 a = int(input())
3 if a > 90 :
4   print('obtuse')
5 elif a == 90 :
6   print('right')
7 else :
8   print('acute')

Writing _0.py
```

Fig. 6 Save the file for the assessment using magic commands.

4) Code Assessment: Code is assessed through the process of code pre-processing, code conversion, execution and error verification, and correct answer verification. Code pre-processing executes code_arrange() defined at the assessment module, removes unnecessary gaps in the assessment code, and checks the input() command in the assessment code to determine whether the input is required or not. Code conversion executes code_convert() defined at the assessment module and adds code to the pre-processed code to output the standard output location from code cell output to a text file. For problems that require input data, the input data in the input() command should be replaced and saved as a Python file. Fig. 7 shows the code that the evaluation code is pre-processed, converted, and stored as a file.

testO.py

```
1 import sys
2 f = open('answer0.txt', 'w')
3 original = sys.stdout
4 sys.stdout = f
5 a = int("1")
6 if a > 90 :
7   print('obtuse')
8 elif a == 90 :
9   print('right')
10 else :
11   print('acute')
12 sys.stdout = original
13 f.close()
14
```

Fig. 7 Example of converted code

Execution and error verification executes the error_check() defined at the assessment module, and if an error occurs, the handling exception provides the student with the code at the code cell output and locations of error occurrence and error cause in their own words, and the evaluation is closed. If this process runs without error, input data is processed by an algorithm and output into the text files. Fig. 8 is an example of outputting a feedback message to code cell output due to an error when executing the converted code.

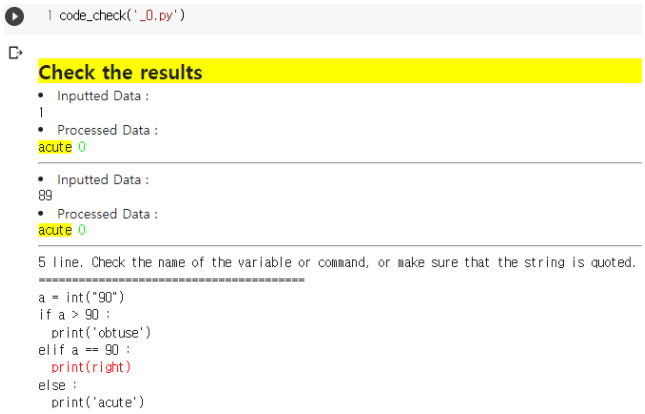

Fig. 8 Feedback when an error occurs

To check the correct answer, the code_test() defined in the assessment module is executed, and the input data, the processed result, and the result are displayed by comparing the processed result to the answer data in the scoring data. The number of scoring data repeats code conversion, execution and error verification, and code assessment, and all scoring data sets are compared to determine the correct answer. Fig. 9 is an example of evaluating the number of scoring data to determine the final correct and wrong answers.

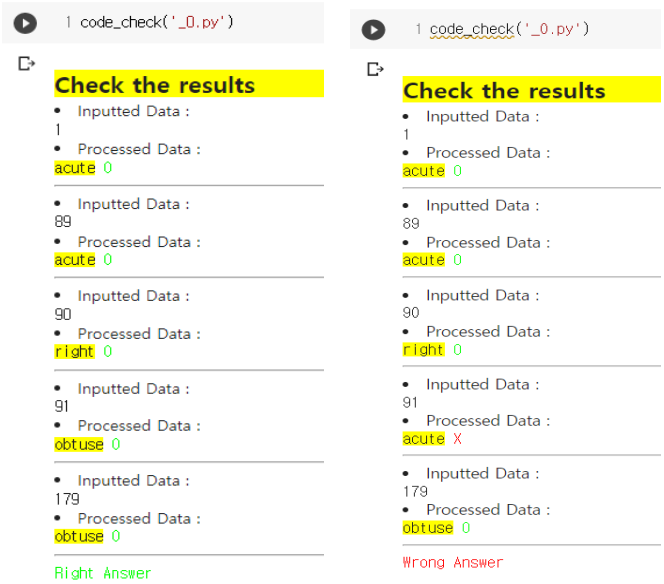

Fig. 9 Example of Correct, incorrect

C. Verification

Scoring data used in other PAA systems were used to verify PAAinJN [30]. Among the items presented in other PAA systems, the 34 problems in four chapters presented in the guidebook [31] and produced based on the curriculum are used for verification.

Chapter 1 is the output, Chapter 2 is the variable and operator, Chapter 3 is the sequential, iterative, and conditional, and Chapter 4 is the array and function, and the number of questions that need and do not need input data by area is shown in Table III. Output, variables and operators, sequential, iterative, conditional, array, and function all operated normally.

TABLE III
OVERVIEW OF DATA TO VALIDATE

Area	Input data ○	Input data X	Total
Chapter1	0	5	5
Chapter2	12	0	12
Chapter3	12	0	12
Chapter4	5	0	5
Total	29	5	34

D. Utilization plan

1) Material development and usage using PAAinJN: Fig. 10 shows how teachers produce teaching and learning materials using PAAinJN and present them to students. The setup is presented to the student by the teacher by writing a setup code, such as in Fig. 3 in the first code cell. The students execute the written code cell and complete the setup. The problem is presented to the student by writing the problem output function and problem variable as a parameter in the code cell. The teacher executes the code and outputs the problem to the code cell output. The student reads the printed question. The teacher writes the assessment code in the code cell, and the magic command and file name(%writefile filename) is presented to the student. The student writes the code from the next line of the magic command and executes it. The code assessment is presented to the student by writing an assessment function and filename as a parameter in the code cell. The student executes the code cell to check whether the code is correct or not.

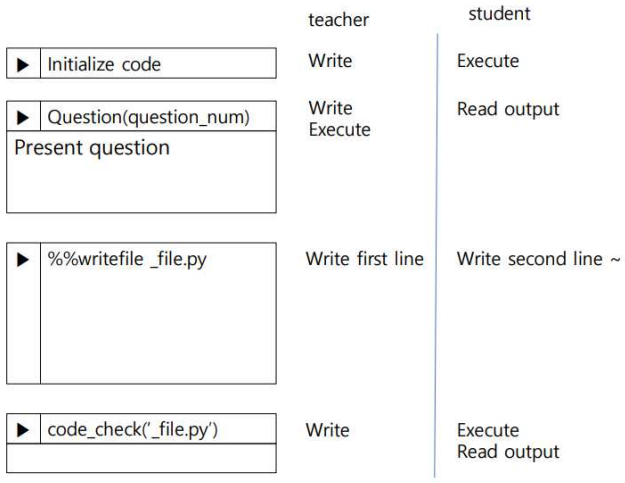

Fig. 10 The role of the teacher when developing materials using PAAinJN

2) Distribution and Collection: Google Colab and Classroom were used to distribute teaching and learning

materials to students using PAAinJN. The classroom is an LMS provided by Google that allows distributing and collecting JN files (.ipynb) in copy form to students. Students can submit the distributed JN files after learning to use Colab provided by Google. Using Colab and Classroom, the teacher can monitor the process of students' learning in real-time [32]. Fig. 11 shows distributing, learning, and collecting data using Classroom and Colab.

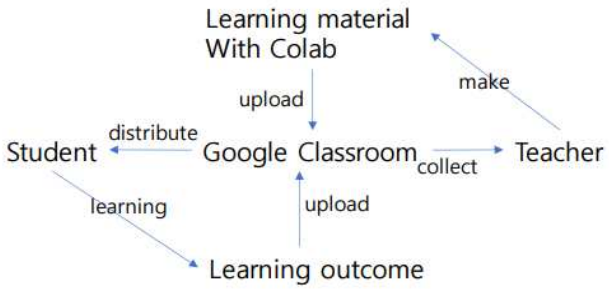

Fig. 11 Process of distributing, learning, and collecting material.

IV. Conclusion

Learning programming is challenging, and various programming tools have been developed to support it. The PAA system and JN support programming learning from a feedback and interactive computing perspective. Combining these features, there were some studies on developing programming tools to utilize the PAA in the JN. However, there were problems with cost due to server operation or SaaS use, difficulties in setting up a server connection and interworking with LMS, and limited interactive computing.

Therefore, in this study, a PAAinJN was designed and implemented to solve the problem of cost and setup and support interactive computing. The PAAinJN develops an assessment module to operate on a personal JN, so there is no server cost. In addition, the assessment module is released on Git and can be used free of charge. The setup for automatic assessment allows the setup to be completed by downloading the assessment module from a code cell instead of creating a config file from another PAA. Problem presentation, assessment code writing, and code evaluation for evaluation were implemented in the form of REPL using code cells and code cell output.

The assessment module includes a problem and consists of a problem module for outputting the problem and a module for assessment of the code. Code assessment is assessed through a code pre-processing process to remove unnecessary gaps and check whether input data is needed, a code conversion process to modify the standard output location and input data, and an execution and error verification process to provide feedback through exception processing in the event of an error.

The setup of downloading and executing an assessment module from a code cell, presents a problem to a code cell output using a problem output function, writing an evaluation code using a magic command and storing it in a kernel, and assessment code using a code assessment function. The implemented PAAinJN verified its performance using examples of other PAA systems' scoring datasets.

The significance of this study is as follows. It does not use servers or SaaS like the design direction so it can be used for free. In addition, it is open to Git, not just to users belonging to a specific group; anyone can use it for free. The setup is completed with two lines of code in the code cell, making it easy to use PAAinJN. In addition, the process of developing learning materials in the JN is the same, and for PAAs, you just add a function defined by the module in the code cell.

The practical limitations of this study and applicable to future studies are as follows. An automatic evaluation is conducted on a personal JN, and the assessment results are presented to the learner. Using LMS, the teacher can collect JN s learned, but data such as submitted codes, evaluation results, and the number of attempts cannot be collected. This has limitations in analyzing learning. Next, there is no problem for PAAinJN. The main feature of PAAinJN is to present the problem to code cell output using a problem output function. If the problem and scoring data sets are included in the problem module, it can be easy to present problems when producing learning materials with the JN. For future research, first, it is necessary to construct a database that can store learning data such as submitted codes, evaluation results, and the number of attempts and to develop a module to transmit them to and from the database. Second, there is a need to develop problems and score datasets linked to the curriculum so that they can be used in school classes.

Acknowledgment

Basic Science Research Program supported this research through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (No. 2019R1I1A3A01060920

References

[1] Lee, D., Hwang, J. Y., Lee, Y., and Kim, S. W, "Informatics and Artificial Intelligence (AI) Education in Korea: Situation Analysis Using the Darmstadt Model". JOIV: International Journal on Informatics Visualization., vol. 6(2), pp. 427-444, 2022, DOI: 10.30630/joiv.6.2.1000.

[2] Shin, S., and Bae, Y, "A Study on the Hierarchical Instructional System Design of Software Education by School System," Journal of The Korean Association of Information Education, vol. 19(4), pp. 533-544, 2015, DOI: 10.14352/jkaie.2015.19.4.533.

[3] Wing, J. M, "Computational thinking," Communications of the ACM, vol. 49(3), pp. 33-35, 2006, DOI:10.1145/1118178.1118215.

[4] Lee, Y. J., Paik, S. H., Shin, J. H., You, H. C., Jeong, I. G., Ahn, S. J., and Jeon, S. G, "Research for introducing computational thinking into primary and secondary education," Korea Foundation for the Advancement of Science and Creativity, BD14060010, 2014.

[5] C. Cachero, P. Barra, S. Meliá and O. López, "Impact of Programming Exposure on the Development of Computational Thinking Capabilities: An Empirical Study," IEEE, vol. 8, pp. 72316-72325, 2020, DOI: 10.1109/ACCESS.2020.2987254.

[6] Kim, J. M, "Problem-solving skills based on computing thinking," Korea Information Processing Society Review, vol. 24(2), pp. 13-21, 2017.

[7] Durak, H. Y., Yilmaz, F. G. K., and Yilmaz, R, "Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities," Contemporary Educational Technology, vol. 10(2), pp. 173-197, 2019, DOI:10.30935/cet.554493.

[8] Malmi, L., Utting, I., and Ko, A, "Tools and Environments," In S. Fincher & A. Robins (Eds.), The Cambridge Handbook of Computing Education Research. Cambridge, United Kingdom, pp. 173-197, 2019, doi:10.1017/9781108654555.022.

[9] Ramírez-Echeverry, J. J., Restrepo-Calle, F., and González, F. A., "A case study in technologyenhanced learning in an introductory computer programming course," Global Journal of Engineering Education, vol. 24(1), pp. 65-71, 2022.

[10] Ullah, Z., Lajis, A., Jamjoom, M., Altalhi, A., Al-Ghamdi, A., and Saleem, F, "The effect of automatic assessment on novice programming: Strengths and limitations of existing systems," Computer Applications in Engineering Education, vol. 26(6), pp. 2328-2341, 2018, DOI: 10.1002/cae.21974.

[11] Kurnia, A., Lim, A., & Cheang, B, "Online judge," Computers & Education, vol 36(4), pp. 299-315, 2001, DOI: 10.1016/S0360-1315(01)00018-5.

[12] Rahmani, A., and Min, J. L, "Automatic Grading System to Supporting Blended Learning in Basic Programming Practice–an Experience Report," In International Conference on Online and Blended Learning 2019 (ICOBL 2019), pp. 96-101, Atlantis Press, DOI: 10.2991/assehr.k.200521.020.

[13] Cardoso, A., Leitão, J. and Teixeira, C, "The Challenges of the Digital Transformation in Education," Proceedings of the 21st International Conference on Interactive Collaborative Learning (ICL2018), vol. 2, pp. 227-236, DOI: 10.1007/978-3-030-11935-5_22.

[14] Wang, J., Kuo, T. Y., Li, L., and Zeller, A, "Assessing and restoring reproducibility of Jupyter notebooks," In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, pp. 138-149, 2020, DOI: 10.1145/3324884.3416585.

[15] Ochkov, V. F., Stevens, A., and Tikhonov, A. I, "Jupyter Notebook, JupyterLab–Integrated Environment for STEM Education," In 2022 VI International Conference on Information Technologies in Engineering Education (Inforino), pp, 1-5, 2022, DOI: 10.1109/Inforino53888.2022.9782924.

[16] Cardoso, A., Leitão, J., and Teixeira, C, "Using the Jupyter Notebook as a Tool to Support the Teaching and Learning Processes in Engineering Courses," In: Auer, M., Tsiatsos, T. (eds) The Challenges of the Digital Transformation in Education. ICL 2018. Advances in Intelligent Systems and Computing, Springer, Cham, vol 917, 2019 DOI: 10.1007/978-3-030-11935-5_22.

[17] González-Carrillo, C. D., Restrepo-Calle, F., Ramírez-Echeverry, J. J., and González, F. A, "Automatic grading tool for jupyter notebooks in artificial intelligence courses," Sustainability, vol. 13(21), pp. 12050, 2021, DOI: 10.3390/su132112050.

[18] Zamfir, F. S. and Pricop, E, "On the design of an interactive automatic Python programming skills assessment system," 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1-5, 2022, DOI: 10.1109/ecai54874.2022.9847414.

[19] Chang, W. Y. and Kim, S. S, "A review on trends of programming(algorithm) automated assessment system and it`s application", The Journal of Korean Association of Computer Education, vol. 20(1), pp. 13-26, 2017.

[20] Han, S., Liu, X., and Woo G., "Evaluation Methods and Classification of Online Judge Systems Based on Education Level and Purpose," KIISE Transactions on Computing Practices, vol.28(10), vol.487-492, 2022, DOI: 10.5626/KTCP.2022.28.10.487.

[21] Ministry of Education, "Informatics Science Guidelines", 2015.

[22] Chang, W. Y., and Kim, S. S, "Development and application of algorithm judging system: analysis of effects on programming learning." The Journal of Korean Association of Computer Education, vol. 17(4), pp. 45-57, 2014.

[23] Jeong, S. S, "The effects of programming education using an automatic programming assessment system on learning flow of general high school students" M.S. thesis, Dept. Computer Education, Graduate School of Korea National University of Education., Cheongju, Korea, 2019.

[24] Johnson, J. W, "Benefits and pitfalls of jupyter notebooks in the classroom," In Proceedings of the 21st Annual Conference on Information Technology Education, pp. 32-37, 2020, DOI: 10.1145/3368308.3415397.

[25] Zhao, P., and Xia, J, "Use JupyterHub to Enhance the Teaching and Learning Efficiency of Programming Related Courses," 23rd International Conference on ISO & TQM, Florida, USA, pp.1-6, 2019.

[26] Klever, N., "Jupyter Notebook, JupyterHub and Nbgrader. Becoming Greener—Digitalization in My Work", The Publication Series of LAB University of Applied Sciences, pp. 37-43, 2020.

[27] Blank, D. S., Bourgin, D., Brown, A., Bussonnier, M., Frederic, J., G ranger, B., and Willing, C, "nbgrader: A tool for creating and grading assignments in the Jupyter Notebook", The Journal of Open Source Education, vol. 2(11), 2019, DOI: 10.21105/jose.00032.

[28] Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Downey, A. B., George, T., ... and Zingale, M., "Teaching and learning with Jupyter," Recuperado: https://jupyter4edu. github. io/jupyter-edu-book, 2019.

[29] Go, H., and Lee, Y, "Exploring how to present the problem of Automatic Assessment system in Jupyter Notebook," Proceedings of the Korean Society of Computer Information Conference, vol. 31(1), pp. 221-222, 2022.

[30] Gyeonggi Science High School Informatics Teachers, Problems. [Online]. available: http://koistudy.net/?mid=viewProblems (accessed: October 1st, 2022).

[31] Han. G. W, Go. G. S, Kim. E. K, Lim. S. K, Jeon. H. S, Jeong. S. S, Jeong. W.Y, Jeong. J. K, Jeong. J. W, and Jo. H. J, "[2015 Revised—-High School] Informatics Guide," Seoul, Korea: SamYangMedia, 2019.

[32] Gupta, A., and Pathania, P., "To study the impact of Google Classroom as a platform of learning and collaboration at the teacher education level," Education and Information Technologies, vol. 26(1), pp. 843-857, 2021.