An Analysis of the Difficulties of Elementary School Students in Python Programming Learning

MiHyun So[#1], JaMee Kim[*2]

Major of Computer Science Education, Graduate School of Education, Korea University,

Anam-ro, Seongbuk-gu, Seoul, 02841, Korea
*E-mail: [#1]brjoa81@korea.ac.kr; [*2]celine@korea.ac.kr*

Abstract— **Software education is spreading all over the world; programming education is being implemented in elementary schools in each country using block-programming language. The block programming language provides a novice environment for easy access, but lacks understanding and teaching all the concepts of programming. There seems to be a need for primary education using text-based programming language for elementary school students. However, the difficulties encountered by learners in typing grammars and commands in textual programming languages could not be completely ruled out. This study aims to investigate the difficulties of learners based on grammatical errors frequently encountered in elementary school students. As a result, the most frequent errors in elementary school students in Python programming were errors caused by missing parentheses, followed by incorrect operator use, assignment of variable values that did not fit the data type, incorrect control statement composition, missing quotes Respectively. This study implies that it provided the improved direction of programming education by identifying the difficulties of learners and presenting practical examples and guidance plans and finding appropriate teaching strategies for text type programming education.**

Keywords— **programming education; python programming error; python**

I. INTRODUCTION

Interest in programming early education is on the rise as the world's education stream recognizes the need for software education. Programming education has already begun in the United States, the United Kingdom, Finland, Estonia, and India starting from elementary schools. It is common in elementary schools in each country to adopt a block-based or visual-based educational programming language that can lessen the cognitive burden of learners[1], [2].

However, as mentioned in the k12 framework in the US in recent years, block programming languages have not removed the conceptual difficulties of learning and teaching programming. Also, misunderstandings about specific concepts such as variables, loops, and Boolean logic have not been extinguished [3]. In order to prevent the misconception of programming due to the block programming language, first programming education using text programming language needs to be done from elementary education.

In the UK's Computing curriculum, people who are between the ages of 11 and 13, they should use two or more programming languages, one of which is the use of text-based programming languages. It also provides guidelines for advanced students in general classes to use their logos and Python programming language to conduct level-specific learning [4].

Research on the potential of Python programming education for elementary school students has been a constant concern for researchers in Korea. Some studies have shown positive results in elementary school students who have completed block-level programming learning, helping to shape algorithms when Python programming training is conducted [5][6]. However, as previous researchers have already pointed out, the difficulties encountered by learners in learning the grammar of textual programming languages and typing commands cannot be completely ruled out[7] - [9].

Therefore, it is necessary to understand the difficulties of elementary school students by analyzing errors that students often make in text-based programming learning for elementary school students. It is expected that educators could introduce text-based programming languages in elementary school primary programming education if they are aware of the mistakes that learners often make, and establish appropriate education strategies for them. In this study, we aim to suggest the improvement of programming education by grasping the difficulties of learners based on errors frequently encountered in elementary school students' Python programming.

II. MATERIAL AND METHOD

A. The Analysis of Programming Errors

A lack of understanding of programming causes programming errors, and the contents of errors repeatedly can be interpreted as difficulties for learners in programming learning. The study of learners' difficulties in programming learning has been an ongoing concern for researchers for a long time. In Korean related studies, elementary school students were found to be error-prone mainly in keeping the grammar rules of the programming language and typing the commands in English. Table 1 summarizes the contents of grammatical mistakes that are frequently made by elementary school students classified by related researchers.

TABLE I
THE CONTENT OF GRAMMAR ERRORS

WaeShik Moon [7]	Data definition and handling errors	Invalid variable assignment
		Inadequate control structure
		Inappropriate control statement structure
	Other errors	Errors caused by inadequate use of the program
		A simple typo
		Hardware failure
Hyeseon Jang [8]	Syntax error	Missing punctuation
		Wrong spelling
		spacing
		Incorrect bracket response
Jaewon Nam Inhwan Yoo [9]	Syntax error	Use unnecessary arguments
		Missing arguments
		Input content that does not match the argument property
		Missing #include statement
		Input semicolon in inappropriate position
		Incorrect use of operators
		The incorrect concept for variables
		Missing task main ()
		Use grammar and wrong control statements
		Incorrect instructions
		Use braces in inappropriate locations
		Variable declaration and value assignment fail
	Coding error	Missing semicolon
		")" Missing parentheses
		"}" Missing braces
		Typo due to mistake

WaeShik Moon [7] classified grammatical errors in VB programming as data definitions, handling errors, and other errors. Types of data definition and handling errors include incorrect variable assignments, inadequate control structures, and incorrect grammar usage. Other errors were categorized as errors due to inadequate use of the program, simple typo, or hardware failure. As a result, the most frequent errors in elementary school students in VB programming were found to be from a simple typo. Next, it was found that the wrong variable was specified with a wrong parameter. Hyeseon Jang [8] classified grammatical errors into punctuation omissions, misspellings, spacing, and erroneous parentheses in Doolittle programming. The results show that there are many errors caused by misspellings, misspellings, and spacing.

Robot programming learning studied by Jae-won Nam and In-hwan Yoo [9] shows the use of additional parameters, missing necessary parameters, and inputting unmatch contents to the parameter attributes. It also shows missing hash (#) which include statements, missing task main (), entering semicolon in inappropriate positions and usage, incorrect concept of variables, misuse grammar, and other control statements. In addition, the inaccurate command input, use brace in inappropriate position, classification of variable declaration and error that could not assign value into grammar error type, missing semicolon and parenthesis, and misplaced typographers were classified as coding errors.

However, missing semicolon and miss the necessary command were analyzed as the most error (paraphrase needed with concise main ideas). As can be seen from the above-mentioned related studies, it can be seen that there has been continuous research on grammatical errors that elementary school students often make. However, it is considered that there is not enough data on the actual cases of how learners are in error and having trouble in programming.

B. Research procedure

This study aims to identify the difficulties that learners experience by analyzing grammatical errors that elementary school students often make when writing programs using Python. The research procedure for achieving some objectives. First, the types of grammatical errors were selected through previous research and literature review. Second, we selected the content based on the elementary information achievement standards of the 2015 revised curriculum and conducted five programming lessons. Third, after analyzing vocabulary program writing test results, we analyzed the error data of learners by collecting the results [10].

C. object of study

This study is composed of a group of after-school students' of W elementary school in Seoul. Nine people are composed of two students in the 4th grade, 2 students in the 5th grade, and 5 students in the 6th grade. Students are students who have completed block programming.

D. Design of lesson

The element of learning in the class was composed of 5th-grade class based on the elementary information achievement standard of the 2015 revised curriculum. Table 2 shows the primary information achievement criteria for programming [11].

| Design a simple program that inputs data, performs the necessary processing, and outputs the results. |
| Understanding the structure of sequential, selection, and iteration in the process of creating a program to solve a problem |

Table 3 shows the contents of each lesson class based on the elementary information achievement standard.

TABLE III
CONTENTS OF EACH CLASS

The First lesson	Learn the characteristics and usage of the Python programming language
The Second lesson	Create a program that calculates your age
The third lesson	Create a Viking boarding program
The fourth lesson	Drawing shapes
The fifth lesson	Find the sum of 1 to 10

In the first lesson, we describe the characteristics and usage of the Python programming language. In the second lesson, we did a programming lesson that uses data input/output, variables, and operators to calculate the age. In the third lesson, the user's key (cm) was input, and the elective programming class was decided to decide whether to ride the rides. In the fourth lesson, the programming learning of various figure drawing using them for a statement of the loop was performed. In the last lesson, a looping programming class was used to find the sum of 1 to 10 using the While statement.

E. Programming Test Questions and Answers

The programming test problem used in this study is to write a vending machine program using variables, input/output of data, and control statements. The programming rules are as follows.

- Create your vending machine (any vending machine can sell anything)
- You can use your control statements such as subsequent statements, selection statements, loop statements, and multiple choice statements.
- The inputs and outputs should be visible.

Figure 1 is a good example of the evaluation criteria.

```
drink=500
money=int(input("please put the money!"))
if money==drink:
    print("Drinking water came out")
elif money>drink:
    print("Drinking water came out")
    print("Change is",money-drink,"won.")
else:
    print("I do not have enought money.")
```

Fig. 1 Good example

III. RESULTS AND DISCUSSION

A. Types and contents of grammar errors

This study selected the types and contents of grammatical errors that occurred in Python programming based on the analysis of previous studies and literature studies [8][10][13]. There are five kinds of classified errors in detail.

TABLE IV
PYTHON GRAMMAR ERROR CONTENT

Type of error	Error contents
Name Error	Spelling error in the command (Case sensitive classification)
Syntax Error	Missing quotes
	Missing parentheses
	Missing period
	Missing colon
	Missing commas
	Incorrect indentation
	Incorrect use of operators
	Incorrect control statement configuration
Attribute Error	Error spelling function or variable (Case sensitive classification)
Type Error	Inappropriate datatype declaration
	Assign a variable value that does not match the data type.
	The missing argument to pass to the function
Value Error	Assigning values that can not be converted to a data type

If you look at the kinds of grammar errors, Name Error is an error that occurs when the command is misspelled or case insensitive. Errors corresponding to Syntax Error include errors due to missing special symbols such as quotes, parentheses opening and closing, periods, colon, and comma, errors due to incorrect notation, incorrect indentation, and incorrect operator usage.

An error that belongs to Attribute Error is an error that occurs when a function or variable is misspelled. Type Error is an error that occurs when declaring a data type that is not appropriate, or by assigning a variable value that does not match the data type, or when the argument passed to the function is missing. Value Error is an error that occurs when a value that cannot be converted is assigned to a data type.

B. Analysis results by type of grammar error

In a program written using Python, grammatical errors frequently encountered by elementary school students were analyzed and Syntax Error was the highest at 79%. Next, Type Error was found to be 21%.
Name Error, Attribute Error; Value Error did not occur. Figure 1 shows the distribution of grammatical errors.
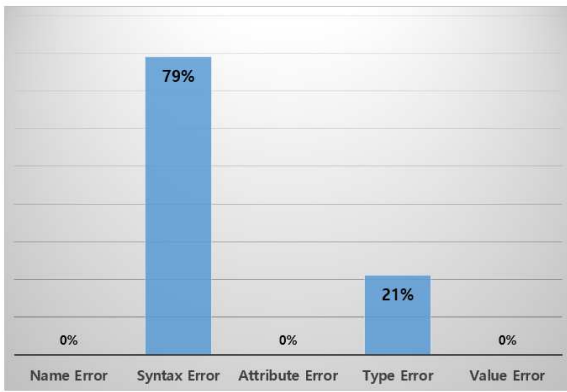
Fig. 2 Error distribution ratio by type

1) Percentage distribution of errors occurred by type:

TABLE V
DISTRIBUTION OF ERROR CONTENT

Type of error	Error contents	ratio
Syntax Error	Missing quotes	7%
	Missing parentheses	36%
	Incorrect use of operators	21%
	Incorrect control statement configuration	14%
Type Error	Assign a variable value that does not match the data type.	21%

In Syntax Error, 36% of errors were caused by missing parentheses, and wrong operators caused 21% of errors. Next, it was found that 14% error occurred and 7% error caused by wrong control statement. In the Type Error, it was found that 21% of the errors were caused by assigning a variable value that does not match the data type.

C. Specific examples of Syntax Error

1) Missing parentheses: In the case of missing *parentheses*, errors often occur when you use the input command to input data to the user with the keyboard and the int command to use the data type conversion. Figure 3 is a partial excerpt of the case where the closing parenthesis of the int command is missing.

```
money=int(input("The Vending machine said, please put the money!")
if money==10000:
    print("The teacher says he ate the pizza")
```

Fig. 3 The case of missing parentheses and causing errors

Figure 4 shows a case where the input command is missing the open/close parenthesis.

```
print("Hello")
print("I'm cell phone vending machine")
money=int(input"please put the money!")
if money<=100:
    print("I'll give you a folder phone.")
```

Fig. 4 Case of missing parentheses and causing errors

There is also a case like a Figure 5 when the parentheses are omitted about the input command.

```
print("please put the money!")
money=input
```

Fig. 5 Case of missing parentheses and causing errors

Based on the above examples, elementary school students may have difficulty coding input () and int () variables in a single line. That is, the process of converting the string input by the user into integer type and storing it in a variable is coded on one line so that the user cannot pay attention to the corresponding parenthesis processing. We could guess that the practice of learning simple commands was done in shell mode. In addition, we could find instances where the input () and int () commands were repositioned, and the int () in the parentheses of input (), and parentheses were missing. The overall reason for making this error is that it is difficult to understand the syntax of int ().

2) Incorrect use of operators: In cases where an error was generated using an invalid operator, errors caused by the assignment operator and the comparison operator were found to be many. Figure 6 is an excerpt of a case where an error occurred using the assignment operator where the comparison operator should be written.

```
print("please put the money!")
money=input
if money=1500:
    print("Drinking water came out.")
```

Fig. 6 Examples of errors using invalid operators

Figure 7 shows an example where the comparison operator misuses the assignment operator and the comparison operatorof the conditional expression is erroneously written.

```
print("please put the money!")
money==input
if money=>500:
    print("Drinking water came out.")
```

Fig. 7 Examples of errors using invalid operators

Based on the above, it can be assumed that elementary school students have difficulty because the comparison operators they had known before are not familiar with grammar rules used as assignment operators in programming.

3) Incorrect control statement configuration: In the case of constructing an incorrect control statement and causing an error, it appears that it occurs in the process of writing multiple if statements. Figure 8 shows that the else statement is misconfigured in multiple if statements and the conditional expression of the elif statement is incorrectly written.

```
if money==49900:
    print("Get Lego Raiders.")
elif money>=49900:
    print("Change is",money-49900,"won.")
else:
    money<49900:
    print("I do not have enought money.")
```

Fig. 8 Examples of errors caused by incorrect control statements

Figure 9 shows a case where an error is generated by erroneously writing an else statement.

```
print("please put the money!")
money==input
if money=>500:
    print("Drinking water came out.")
else money<500:
    print("I do not have enought money.")
```

Fig. 9 Examples of errors caused by incorrect control statements

Based on the above examples, it can be inferred that elementary school students have difficulty in understanding the basic structure of an if statement. Especially, students with lower grades showed difficulty in constructing if statements.

D. Specific examples of Type Error

1) Assign a variable value that does not match the data type: Figure 10 shows an example of a case where an error occurs by assigning a variable value that does not match the data type.

```
money=input("The amount is 49900 won. please put the money!")
if money==49900:
    print("Get Lego Raiders.")
```

Fig. 10 An error occurred by assigning a variable value that does not match the data type

Figure 10 shows a case where an error occurs because input value is not converted to integer type by input (). This was also common for students at lower grade levels. Based on the above examples, it can be assumed that the lower the grade level, the more difficult the input () grammar becomes.

IV. CONCLUSION

This study analyzed the grammatical errors frequently encountered by elementary school students in Python programming learning and identified the difficulties that learners face. Research shows that missing parentheses most often cause grammatical errors in elementary school students in Python programming, followed by incorrect operator usage, assignment of variable values that do not fit the data type, incorrect control statement composition, missing quotes Respectively.

In cases where parentheses are missing, it is often the case that one of the corresponding parentheses is missing or the parentheses following the function are missing. The process of converting the input string to an integer, i.e., int () and input () and that this difficulty is experienced. For high-grade learners, extra care must be taken to ensure that the corresponding parentheses are not omitted, and low-grade

learners may need to understand the grammar of input () correctly.

Second, when we look at the cases in which elementary students made mistakes using the wrong operators, the comparison operators and the assignment operators confused most of them. This seems likely to be improved through sufficient practice and constant teacher attention.

Third, when we look at the cases where an error is caused by assigning a variable value that does not fit the data type, it is often seen that the variable value of the input string is incorrectly assigned to the formula. Especially, the lower the grade, the higher the tendency. It seems necessary to understand the grammar of input () correctly because of the difficulty of understanding the grammar of input (). In addition, it is necessary to go through the edit mode environment rather than the shell mode when conducting the class, and repeated practice is needed to understand the data types.

This study aims to identify the grammatical mistakes that elementary students frequently make and to provide educators with the direction of improved programming education aimed at improving the problem-solving ability of learners by seeking appropriate teaching strategies for text type programming education It has significance.

Fourth, most of the cases in which an error is made by constructing an incorrect control statement often occur in the process of writing an if statement. Mainly, it was found that the less learner had difficulty in writing the conditional expression in the else statement and the lack of understanding about elif and else. It is necessary to teach the elif statements to be learned after the preliminary learning of the basic structure of if and else is made sure [14].

REFERENCES

[1] S. K. Shin, Y. G. Bae, "A Study on the Hierarchical Instructional System Design of Software Education by School System", *The Journal of The Korean Association of information Education, vol. 19, pp.533-544, December.2015*

[2] S. J. Lee, J. M. Kim, and W.G. Lee, "Analysis of Factors Affecting Achievement in Maker Programming Education in the Age of Wireless Communication", *Wireless Personal Communications, vol.93, pp.187-209,2016*

[3] *k12 computer science frame work, K12, 2016.*

[4] Department for Education (2013). National curriculum for England: Computing programmes of study. London, England: Department for Education.

[5] M. H. So, J. M. Kim, "Transference from learning block type programming to learning text type programming", *The Journal of Korean association of computer education, vol. 19, pp.55-68, november. 2016.*

[6] S. S. Kim, "Design model for Python programming class utilizing scratch : focused on fifth grade elementary school"

[7] J. S, Kim, Y, S, Kim, "Analysis of Learner`s Characteristics and Relationship between Learning Styles and Achievements in Online Programming Course", *The Journal of Korean association of computer education, vol. 18, pp.59-68, May. 2015.*

[8] W. S. Moon, "Analysis of Error Types occurring on Elementary School Student's Programming Learning", *The Journal of Korean association of computer education, vol. 11, pp.319-327, May. 2006.*

[9] H. S. Jang, "Design of Error Feedback System in Programming Learning for Novice", Ph. D. Edu. thesis, University of Korea, Seoul, Korea, 2007.

[10] D. Hooshyar, R. B. Ahmad, M. Yousefi, M. Fathi, S. J. Horng, and H. S. Lim, "SITS: a solution-based intelligent tutoring system for students' acquisition of problem-solving skills in computer programming", *Innovations in Education and Teaching International*, DOI:10.1080/14703297.2016.1189346

[11] Informatics Curriculum, Ministry of Education, Republic of Korea. 2015-74, 2015.

[12] J. W. Nam, "Development of NXC Robot Programming Supporting System Based on Types of Programming Error", *The Journal of The Korean Association of information Education, vol 15.pp.375-385, september.2011*

[13] S. C. Lee, *Python for everyone, 4nd ed., Korea: Gilbut, 2017.*

[14] R. Jung, "Analysis of characteristics of novice programmers and teaching schemes", *Theses collection, Samchok National University*, vol. 29, 1996.