論文

Programming Knowledge of Japanese High School Graduates

Christopher PIROTTO[*1], Daniel COOK[*1], and Masahiro OSOGAMI[*2]

[*1] Organization for Fundamental Education, [*2] Department of Management and Information Science

This paper investigates Japanese high school graduates' programming knowledge, specifically those who have enrolled in a university major requiring multiple university-level programming courses. Participants were surveyed about their programming knowledge, general interest in computers, access to computers, and computer usage. Questionnaire data was analyzed, revealing low student computer usage despite high interest in computers and programming. Furthermore, concerns arise regarding limited student access to computers outside of the school system. Results show that only 82 of the 173 students had programming experience, but many of these students struggled to identify programming languages they reported to be familiar with, raising doubts about their experience. These findings offer insights into the programming and computer knowledge of Japanese high school graduates and highlight challenges faced by university professors in preparing students for IT careers.

Key Words : Programming Knowledge, High School Programming Education, Personal Computing Competency

1. Introduction

Programming is crucial in today's technology-driven world, particularly for job seekers in the information technology (IT) field, and Japan is no exception. The Japanese government has recognized the need for more IT engineers to meet the growing demands of the technology workforce (Government of Japan, 2013). Consequently, policy changes have been implemented, including the introduction of programming curriculum in primary and secondary schools (Kanemune, 2019). Universities in Japan have also responded by offering new courses of study like data science to address the increasing needs of the IT sector.

However, concerns persist regarding the programming knowledge and basic computer skills of Japanese high school graduates, raising doubts about their readiness for IT-related majors at the university level and their understanding of what a career in IT may entail. This study aims to examine the current state of computer usage and programming knowledge among Japanese high school graduates. The authors hope that the findings will contribute to the development of policies and programs that improve computing skills and programming knowledge among Japanese students.

Therefore, this study has the following four research goals:

① Determine the number of students with programming experience, their frequency of computer use, and assess their basic computer proficiency based on self-reported typing scores.
② Assess the level of interest in computers and programming among high school graduates.
③ Explore the nature of programming experience among students with programming experience.
④ Identify reasons for the lack of experience with computers or programming.

2. Programming Curriculum in Japanese Schools

In the past decade, there has been a global interest to integrate programming into educational curricula, including in Japan. This is driven by the high demand for IT engineers in the rapidly expanding technology job sector. This section examines the current state of programming curriculum in Japanese schools and provides an overview of its development.

---

* 原稿受付　2023 年 5 月 1 日
*1 基盤教育機構
E-mail: chris-pirotto@fukui-ut.ac.jp

Following Japan's announcement in 2013 to address the workforce demands of the IT sector, curriculum changes were initiated to nurture the next generation of Japanese youth. In 2020, the Ministry of Education implemented mandatory programming classes at primary, junior high, and high schools (Kanemune, 2019). Primary school courses primarily focus on introducing programming concepts through graphical user interface (GUI) environments like Scratch, emphasizing programming logic before introducing character user interface (CUI) environments. Middle school students learn about the integration of programming with hardware, while high school students are introduced to problem-solving with algorithms.

However, there are significant concerns regarding the implementation of the new programming curriculum and challenges associated with its execution. Nakamura (2021) highlighted issues such as inadequate funding, a shortage of knowledgeable faculty members, and large class sizes. These concerns align with the speculations made by Kanemune et al. (2017) when the new curriculum was announced, suggesting that most teachers lack IT training and that training enough teachers before implantation is nearly impossible. Nishikawa et al. (2021) observed the new curriculum in action and interviewed teachers responsible for its implementation, identifying challenges related to lesson planning, computer and programming environment preparation, and methods for evaluating student progress. They concluded that the Ministry of Education, boards of education, and administrators need to provide more resources and training to successfully implement the new programming curriculum.

While many countries have implemented computer science related curricula, there is limited information on the effectiveness and outcomes of these programs, making it difficult to compare the programming knowledge and computer literacy levels of Japanese high school graduates on a global scale (Vegas and Fowler, 2020).

3. Study Design

3.1 Instrument

A questionnaire was developed by the researchers to examine the programming experience, knowledge, and interest of recent Japanese high school graduates. The questionnaire also included inquiries about computer access, computer usage, and several open-ended questions. The Japanese version of the questionnaire can be made available upon request.

The first section of the questionnaire assesses students' general interest in computers using a Likert scale with five items. It also includes questions about typing confidence and computer accessibility. Based on the responses to a question regarding programming experience, participants who indicated no prior exposure to programming were asked to provide brief explanations for this lack of experience. Those who reported having used or studied programming at least once were presented with various programming-related questions to evaluate their experience and knowledge. Furthermore, participants were tested on their understanding of programming concepts through Python-specific programming questions, as Python is particularly relevant to the readiness of high school graduates for university-level programming courses.

3.2 Participants

The participants of this study were first-year students at a rural private university in Japan. The survey was administered to all students during the first week of classes, about one month after their high school graduation. Since the students had not yet commenced their university curriculum, they were suitable for assessing the programming knowledge of high school graduates.

Although all participants attended the same university, they were enrolled in two different departments: The Department of Management and Information Science (referred to as Information Science) and the Department of Electrical, Electronic, and Computer Engineering (referred to as Electrical Engineering). These majors were selected because all students pursuing these majors are required to complete at least one year of programming courses. The Information Science students' courses primarily focus on Python, which prepares them for careers in areas like data science, logistics, and business. The Electrical Engineering students, on the other hand, are required to learn C and Python for one semester each to equip them for roles as electronic engineers in fields such as technology, information, and space. A total of 173 students participated in this study, with 97 from the Department of Management and Information Science and 76 from the Department of Electrical, Electronic, and Computer Engineering. No personal information, including gender, was collected from the participants.

3.3 Data Collection and Analysis

The questionnaires were administered to students during regular class hours, and it was emphasized that participation was optional and would not affect their course grades. The questionnaires were conducted in Japanese, the native language of most participants and the primary language at the university. Students who chose to participate were provided with a QR code that directed them to a Google form. In this form, students were informed about the purpose of the questionnaire, the voluntary nature of participation, and the contact information of the researchers. The collected data was automatically imported into Google Sheets for data cleaning. Some open-ended responses consisted of “I don’t know” or “Nothing to share” and these responses were excluded from the results section of this study. Descriptive statistics were primarily used to analyze the data, although *t*-tests and ANOVA analyses were employed at times to compare the mean averages of different student groups.

4. Results

This section describes the findings related to the four research goals, with each goal addressed in a dedicated subsection. The first subsection examines the number of students with programming experience, their computer usage, and typing proficiency. Following that, the level of interest in computers and programming is reported. The third subsection focuses on students with programming experience, providing insight into the type and extent of their programming experience. Conversely, the final subsection discusses the reasons provided by inexperienced students for not having learned programming yet.

4.1 Programming Experience, Computer Usage, and Typing Proficiency

The initial set of questionnaire items aimed to assess students’ exposure to programming, their general computer usage, and their typing proficiency. This provided a broad understanding of their computer usage, proficiency, and experience. The first item, presented in Table 1, provides a breakdown of the number of students who have learned, attempted, or studied programming at least once in their life. More than half of the students (52.6%, $n = 91$) had not engaged in programming, while 47.4% of students ($n = 82$) had some experience with programming. Analyzing the responses based on department, the Information Science students leaned more towards “No”, with 54.6% ($n = 53$) reporting no programming experience and 45.4% ($n = 44$) reporting some experience. The responses from Electrical Engineering students were evenly split, at 38 students each.

Table 1: Classification of Students Based on Programming Experience

Department	Have you studied, learned, or attempted to do programming, even if just once?		Total
	Yes	No	
Information Science	44	53	97
Electrical Engineering	38	38	76
Total	82	91	173

The subsequent question aimed to provide a general overview of students’ computer usage frequency, even for basic tasks such as internet browsing, gaming, completing assignments, or watching YouTube videos. Figure 1, a 2D stacked bar chart, illustrates the distribution of computer usage frequency for all students. According to Figure 1, the majority of students use computers at least a few times a week. However, 14.5% ($n = 25$) of respondents reported using a computer hardly ever. Among these students, the majority ($n = 20$) were from the Information Science major while only 5 were from Electrical Engineering. It is worth noting that only 32.9% ($n = 57$) of all students reported using a computer daily. In contrast, 67.1% (n = 116) of students use a computer a few times a week or less frequently. This finding is significant considering that their future careers in their respective majors will likely require daily computer usage.

The students were also asked to rate their typing ability on a scale from one to ten, with ten indicating high proficiency and one indicating low proficiency. While modern user interfaces and websites are designed to be mouse-centric, it is important to note that typing proficiency is relevant to programming and related fields. Table 2 presents the average self-rated typing scores based on programming experience and major. Table 3 shows the average self-reported typing score cross-referenced by frequency of computer usage and major.

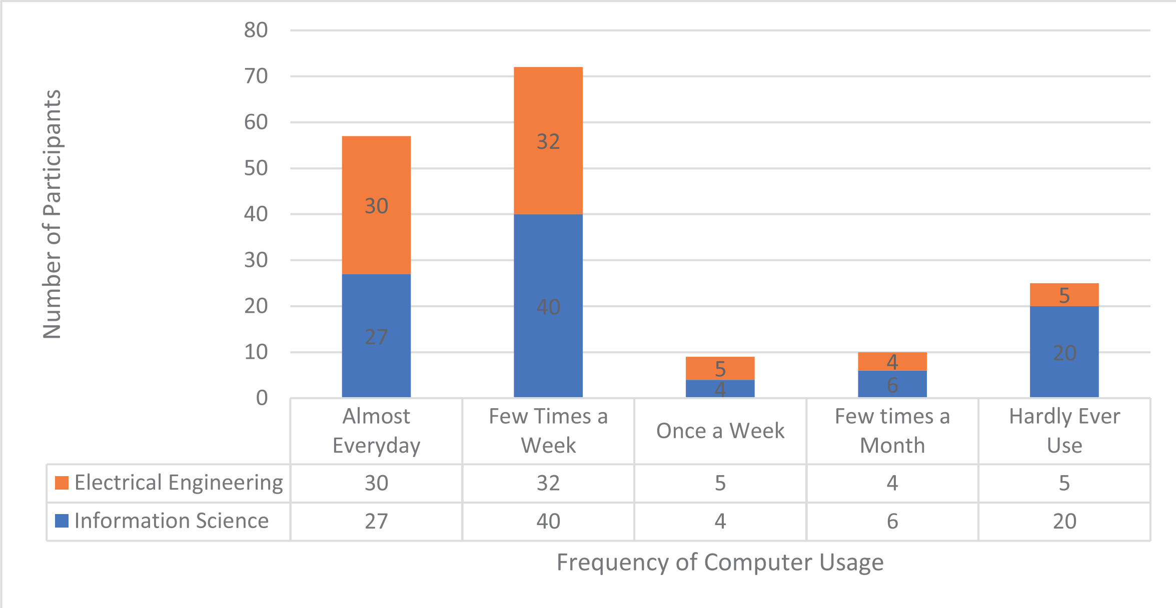

Fig. 1: Frequency Distribution of Computer Usage among all 173 Participants - 2D Stacked Bar Chart

Table 2: Comparison of Mean Self-Reported Typing Ability by Programming Experience and Department

Department	**How do you rate your ability to type on a computer from 1 (not proficient at all) to 10 (very proficient)?**		**Total**
	Programming Experience	**No Programming Experience**	
Information Science	5.43	3.32	4.28
Electrical Engineering	5.50	3.84	4.67
Total	5.46	3.54	4.45

Table 2 clearly demonstrates a significant difference in self-reported typing ability based on whether students have previous programming experience or not, regardless of their major. This difference was statistically significant for all students combined ($t(171) = 6.18$, $p < .001$, 95% CI [1.31, 2.54]), Information Science students ($t(95) = 5.11$, $p < .001$, 95% CI [1.29, 2.93]), and Electrical Engineering students ($t(74) = 3.48$, $p < .001$, 95% CI [0.71, 2.61]).

Regarding Table 3, it is not surprising to find that the more frequently students use a computer, the more confident they are in their typing ability. To confirm the statistical significance of these results, a one-way between groups ANOVA was conducted across all participants regardless of major. Group 1 consisted of participants who reported using a computer "Almost Everyday" (mean = 5.46, $SD = 2.35$, $n = 57$), Group 2 included those who used a computer "A few times a week" (mean = 4.46, $SD = 2.05$, $n = 72$), and Group 3 consisted of participants who reported computer usage as "Once a week, "A few times a month", or "Hardly ever" (mean = 3.14, $SD = 2.35$, $n = 44$).

The one-way ANOVA revealed a statistically significant difference in mean self-reported typing proficiency between at least two groups ($F(2, 170) = 15.33$, $p < .001$). Further *t*-tests were conducted for multiple comparisons, revealing statistically significant differences in means between Group 1 and Group 2 ($t(127) = 2.58$, $p = .011$, 95% CI [0.23, 1.76]), Group 2 and Group 3 ($t(114) = 3.54$, $p < .001$, 95% CI [0.58, 2.06]), and Group 1 and Group 3 ($t(99) = 5.46$, $p < .001$, 95% CI [1.48, 3.16]). These results confirm that the frequency of computer usage has a direct positive impact on students' self-reported typing ability, supporting the intuitive expectation.

Table 3: Comparison of Mean Student Self-Reported Typing Ability by Computer Usage and Major

Computer Usage Frequency	How do you rate your ability to type on a computer from 1 (not proficient at all) to 10 (very proficient)?		Total
	Information Science	Electrical Engineering	
Almost everyday	5.37	5.53	5.46
Few times a week	4.43	4.50	4.46
Once a week	4.25	3.40	3.78
Few times a month	4.17	3.75	4.00
Hardly ever use	2.55	2.60	2.56
Total	4.28	4.67	4.45

4.2 Interest in Computers and Programming

Table 4 presents the mean scores of five Likert-item questions that aimed to measure the interest of high school graduates in various aspects related to computers and programming. The questions asked participants to rate their interest on a scale of one to ten, with one indicating no interest and ten indicating extreme interest. The five items assessed interest in programming, computer hardware, computer-assisted learning, things that can be done with computers, and how IT companies' computer usage. The average of the five items represented overall computer interest. The mean scores for each are reported in Table 4.

Table 4: Mean Student Interest Levels in Various Aspects of Computers

Interest in	Department		Total
	Information Science	Electrical Engineering	
Programming	6.38	7.91	7.05
Hardware	5.77	7.29	6.44
How PCs help you learn	5.95	6.88	6.36
Things you can do with PCs	6.44	7.57	6.94
How IT companies use PCs	6.67	7.07	6.84
Overall computer interest	6.24	7.34	6.73

Table 4 displays the students' high overall interest in computers. The scale ranged from 1 (no interest) to 10 (extreme interest), with a neutral score of 5.5. Both majors expressed considerable interest in all computer aspects examined. Information science majors showed relative indifference towards computer hardware and generally lower interest across all items compared to Electrical Engineering majors. Electrical Engineering majors demonstrated the highest interest in programming, while Information Science majors displayed greatest interest in IT companies' computer usage.

Table 5 presents correlation coefficients (r) among the 5 Likert items assessing computer interest, indicating a strong association overall. While Item 5 demonstrates the weakest associations with correlation coefficients of $r = 0.59$ (Item 2) and $r = 0.62$ (Item 1), these values still fall within the category of moderate to strong.

Table 5: Correlation Matrix of 5 Likert-Items Measuring Student Interest in Computers

Interest in	Programming	Hardware	How PCs help you learn	Things you can do with PCs	How IT companies use PCs
Programming	1.00	0.80	0.71	0.75	0.62
Hardware	***	1.00	0.74	0.78	0.59
How PCs help you learn	***	***	1.00	0.75	0.64
Things you can do with PCs	***	***	***	1.00	0.67
How IT companies use PCs	***	***	***	***	1.00

Additionally, Table 6 presents the findings of an exploratory factor analysis with Promax rotation and reveals high factor loading scores for all factors, including Item 5 with a score of 0.72. These findings suggest that the five items can be loaded onto a single factor to measure overall computer interest, accounting for 70.8% of the variance. The results of the correlation coefficients and factor loadings indicate that the five items effectively capture overall computer interest. Examining the overall computer interest of students in Table 4, it is evident that the students express a very high level of interest in computers. However, it is also important to identify subsets of students who report lower levels of interest.

Table 7 presents overall computer interest based on frequency of computer usage. It reveals a correlation between decreased computer frequency and reduced overall interest in computers. Two instances deviate from expectations: Information Science students who use computers “Once a week” reported a score of 6.85, and Electrical Engineering students who use computers “A few times a month” reported a score of 7.25. This can be explained by each group containing only four participants. Given the limited number of participants in certain groups, an ANOVA comparison between groups was not conducted. Descriptive statistics alone demonstrate that as computer usage frequency increases, so does overall interest in computers.

Table 6: Factor Loadings for 5 Likert-Items from Exploratory Factor Analysis with Promax Rotation

Interest in	Programming	Hardware	How PCs help you learn	Things you can do with PCs	How IT companies use PCs
Factor Loadings	0.87	0.89	0.84	0.88	0.72

Table 7: Mean Overall Computer Interest based on Frequency of Computer Usage

Computer usage frequency	Overall computer interest (1 = not interested at all, 10 = Extremely interested)		Total
	Information Science	Electrical Engineering	
Almost everyday	6.81	7.88	7.38
Few times a week	6.63	7.09	6.84
Once a week	6.85	6.84	6.84
Few times a month	5.2	7.25	6.02
Hardly ever use	4.89	6.28	5.17
Total	6.24	7.34	6.73

Table 8 presents mean computer interest based on whether students have tried, learned, or studied programming at least once. One notable finding, as expected, is that students with programming experience exhibit higher interest in computers compared to those without programming experience ($t(171) = 4.71$, $p < .001$, 95% CI [0.76, 1.84]). Further analysis by major reveals statistically significant results for Information Science students ($t(95) = 3.79$, $p < .001$, 95% CI [0.70, 2.22]) and Electrical Engineering students ($t(74) = 2.80$, $p = .006$, 95% CI [0.29, 1.69]). These findings shed light on the overall computer and programming interest among students. Both majors require extensive computer usage, at least a year of dedicated programming learning, and preparation for careers where computer skills are essential. Despite these requirements, there exists a subset of students who display indifference towards computers and possess limited computer experience.

Table 8: Mean Overall Computer Interest based on Programming Experience

Programming Experience	Overall Computer Interest (1 = not interested at all, 10 = Extremely interested)		Total
	Information Science	Electrical Engineering	
Yes	7.05	7.84	7.41
No	5.58	6.85	6.11
Total	6.24	7.34	6.73

4.3 Type and Level of Experience

The third research question specifically targeted students who reported having learned, studied, or attempted programming at least once. As previously reported in Table 1, this accounted for 82 of the total 173 participants. The questionnaire included items to determine the sources of their programming experience, the programming languages they were familiar with, and the extent of their experience. Regarding the source of their programming experience, students were asked three yes-or-no questions. The first question inquired whether they learned programming in their junior high school classes, the second asked about programming education in their high school classes, and the final question assessed whether they acquired programming experience through self-study. Participants were allowed to provide multiple positive responses, and they answered each question independently. The percentage of "Yes" responses is presented in Table 9 below. Table 10 displays the number of positive respondents relative to the total number of participants in the study (173).

Table 9: Breakdown of Sources of Programming Experience among Students with Programming Experience (*n* = 82)

Major	Programming Experience Location		
	Junior High	High School	Self-Study
Information Science (*n* = 44)	13	35	2
Electrical Engineering (*n* = 38)	9	30	14
Total (n = 82)	22	65	16

As shown in Table 9, 22 of the 82 students with programming experience learned programming in their junior high school classes. This percentage remains consistent across both majors. The number of students who acquired programming knowledge in their high school classes significantly increased to 65 of the 82 students (79.27%), with similar percentages observed for each major. However, a notable disparity arises when examining self-study experiences. Among Information Science majors, only 2 out of 44 participants (4.54%) reported self-studying programming, while 14 out of 38 Electrical Engineering students (36.84%) reported the same. It is important to note that these percentages reflect the participants with programming experience and do not represent the entire study cohort.

Participants were also queried about the number of programming projects they had undertaken that had more than 100 and 1000 lines of code. These corresponding findings are presented in Table 10.

Table 10: Number of Projects Students (*n* = 82) have Made with over 100 or 1000 Lines of Code

Number of projects	Numbers of students 100 lines of code			Number of students 1000 lines of code		
	Information Science	Electrical Engineering	Total	Information Science	Electrical Engineering	Total
0	38	21	59	42	33	75
1-2	5	11	16	2	5	7
3-5	0	6	6	0	0	0
More than 5	1	0	1	0	0	0

According to Table 10, The majority of students from both departments reported never having written a program or worked on a project exceeding 100 lines of code. Out of the 82 students with programming experience, only 23 had such experience, 17 belonged to the Electrical Engineering major. Furthermore, only 7 students had experience with projects exceeding 1000 lines of code, 5 of them from the Electrical Engineering major.

Participants who reported programming experience were surveyed regarding their familiarity and experience with various programming languages. Figure 2 displays the participants' responses. Scratch, despite not being a programming language, was included in the survey due to its usage in Japanese primary schools. The total number of participants with programming experience was 82, comprising 44 from the Information Science major and 38 students from the Electrical Engineering major. Participants were allowed to select multiple programming languages or specify ones not listed in the survey.

Programming Language	Have Heard of			Have Experience With		
	Information Science (n = 44)	Electrical Engineering (n = 38)	Total (n = 82)	Information Science (n = 44)	Electrical Engineering (n = 38)	Total (n = 82)
Java	77.27%	73.68%	75.61%	9.09%	26.32%	17.07%
C	68.18%	68.42%	68.29%	25.00%	50%	36.59%
Python	72.72%	60.53%	67.07%	34.09%	28.95%	31.71%
JavaScript	36.36%	52.63%	43.90%	4.55%	21.05%	12.20%
C++	31.81%	57.89%	43.90%	9.09%	18.42%	13.41%
C#	22.72%	36.84%	29.27%	2.27%	7.89%	4.88%
Swift	6.82%	13.16%	9.76%	0%	2.63%	1.22%
Scratch	4.54%	10.53%	7.32%	2.27%	5.26%	3.66%
Scala	2.27%	7.89%	4.88%	0%	0%	0%
Lua	2.27%	5.26%	3.66%	0%	0%	0%

Fig. 2: Heatmap of Programming Languages Known and Used by Students with Programming Experience

As seen in Figure 2, there are three tiers of programming languages students have heard of. Tier 1 includes Java, C, and Python, with a total percentage of 60-77% across majors. Tier 2 comprises JavaScript, C++, and C#, scoring between 22-57%. A few students in each major have heard of Swift, Scala, and Lua. couple of students in each major had heard of Swift, Scratch, Scala, and Lua. C and Python have the highest experience ratings. C is the most experienced language, with 50% of Electrical Engineering students and 25% of Information Science students having experience. Python follows closely, with 34% of Information Science students and almost 29% of Electrical Engineering students having first-hand experience.

To assess students' proficiency and knowledge in the programming languages they claimed experience with two types of questions were asked. The first set presented "Hello World" code snippets from Java, JavaScript, Python, C, and C++. Students were asked to identify the programming language of each code snippet or indicate they did not know. A "Hello World" program is traditionally the first program a beginner will learn, consisting only of the necessary code to print the text "Hello World". Table 11 shows the number of students with programming experience who correctly identified the languages.

Table 11: Number of Students Able to Correctly Identify the Programming Language of the Hello World Program

Programming language	Number of students who correctly identified the language		Total ($n = 82$)
	Information Science ($n = 44$)	Electrical Engineering ($n = 38$)	
Java	1	2	3
C	4	9	13
Python	6	12	18
JavaScript	0	3	3
C++	1	2	3

Out of the 26 students who claimed to have used Python, only 18 correctly identified a "Hello World" program. Similarly, out of the 30 students who reported using the C programming language, only 13 students were able to correctly identify it.

To assess the extent of participants' programming knowledge, a series of Python questions were presented. The first six questions were multiple-choice and focused on programming logic and Python-specific knowledge. The remaining four questions required short answers, asking students to write the output of given programs. Table 12 provides a summary of the results for the ten questions, including labels indicating the programming concepts being tested.

Table 12: Number of Students Able to Correctly Answer Each Programming Question

Programming concept	Number of students who answered the specific concept correctly		Total
	Information Science (*n* = 44)	Electrical Engineering (*n* = 38)	(*n* = 82)
1. Concatenate Multiple Strings	23	19	42
2. Remainder Operator	6	3	9
3. Order of Operations and Floats	4	2	6
4. Python Reserved Words	19	13	32
5. if-elif-else Conditionals #1	6	5	11
6. if-elif-else Conditionals #2	6	9	15
7. Functions and Conditionals	1	3	4
8. Object Creation and Usage	0	0	0
9. String Manipulation	0	0	0
10. Regular Expressions	0	0	0

More than half of the students demonstrated a correct understanding of string concatenation in Python, while 32 out of 82 students accurately identified non-reserved words in Python. A small subset of students (11 for question 5 and 15 for question 6) exhibited comprehension of the logic behind if-elif-else conditional statements. However, a low percentage of students possessed knowledge of Question 2 (Remainder Operator) and Question 3 (Order of Operations and Floats), which required Python-specific understanding of number types and operators. Notably, only 4 out of 82 students grasped the concept of functions, and no students answered intermediate-level questions related to objects, string manipulation, or regular expressions. These findings suggest that students' programming experience and knowledge, particularly in Python, is very basic.

Comparing the results of the "Hello World" questions and the initial Python-specific questions reveals certain disparities. While only 18 out of 82 students correctly identified the Python "Hello World" program, 42 out of 82 students accurately responded to the question about string concatenation in Python. Furthermore, 32 out of 82 students understood the concept of Python reserved words. Several factors may account for these discrepancies. One factor could be the intuitiveness of the Python-specific questions. Additionally, the multiple-choice format of these questions, where the correct answer was among the provided options, may have facilitated higher accuracy compared to open-ended questions. Moreover, students could have made lucky guesses without truly comprehending the question or answer choices. It is also worth considering that some results from the Python-specific questions might be influenced by crossover effects. Many programming languages share common syntax and keywords, allowing students to understand a code snippet in another language without practical experience.

Previously, it was stated that 34% of Information Science students and nearly 29% of Electrical Engineering students claimed to have experience with Python. However, examining the reported experience alongside the results of the Python-specific questions suggests that students' actual experience is quite limited.

4.4 Reasons for Inexperience

The fourth objective of the research aimed to uncover potential factors contributing to participants' lack of programming or computer experience. All participants were asked questions regarding their computer access, and specifically, those who indicated no programming experience were asked an open-ended question to explore the primary reasons behind their non-engagement with programming. All 173 participants were asked the following questions: 1) Where did you primarily use a computer during high school? 2) Did you have your own computer in high school? 3) How many computers did your family own while you were in high school? Tables 13 to 15 present the results of these inquiries.

Table 13 indicates that the majority of students did not possess their own computer during high school. While approximately 50% of Electrical Engineering students had personal computer access, this figure dropped to around 24% for Information Science students. Notably, Table 14 reveals that 7% of the Information Science students did not have a computer in their house while in high school, and nearly half of them had only one computer shared among all family members. Although to a lesser extent, 30% of Electrical Engineering students also shared a single computer with their household members.

Table 13: Participants' Ownership of a Personal Computer during High School

Did you have your own computer in high school?	Department		
	Information Science (*n* = 97)	Electrical Engineering (*n* = 76)	Total (*n* = 173)
Yes	23	37	60
No	74	39	113

Table 14: Number of Computers Present in Participants' Households during High School

How many computers did you have in your house while in high school?	Department		Total (n = 173)
	Information Science (*n* = 97)	Electrical Engineering (*n* = 76)	
0	7	1	8
1	45	23	68
2	26	30	56
3	12	15	27
4 or more	7	7	14

Considering that 44% of participants either lacked access to a computer at home or had to share one among family members, the responses to the third question in Table 15 align with expectation. Out of 173 participants, 101 reported using a computer primarily at school. While slightly over half of Electrical Engineering students indicated using a computer primarily at home, 46% stated that school was their primary location. In contrast, Information Science students displayed a greater tendency to use computers at school, with 68% reporting this while only 28% reported using a computer primarily at home.

Table 15: Primary Location of Computer Usage among Participants during High School

Where did you primarily use a computer?	Department		Total (*n* = 173)
	Information Science (*n* = 97)	Electrical Engineering (*n* = 76)	
Home	27	40	67
School	66	35	101
Library or cram school	0	1	1
Didn't use a computer	4	0	4

Tables 13 to 15 provide valuable insights into the accessibility challenges that high school students encounter outside of school when it comes to computer access. These findings are further reinforced by the responses to the final open-ended question posed to students without programming experience. Participants were asked to identify the primary reason for not having learned, attempted, or studied programming thus far, and Table 16 presents the results.

Table 16 shows the closest translations of participants' responses in Japanese. The most expressed phrase was "no chance". While the exact meaning behind this vague response is not clear, it could indicate that students perceive programming opportunities as something that should be provided to them rather than actively seeking out those opportunities themselves. For Information Science students, the major obstacles to learning programming include a lack of opportunities to learn, programming not being taught in school, and a lack of personal interest. Conversely, Electrical Engineering students provided more specific answers such as "Insufficient time" or "My home environment" or "Lack of access to a computer". The responses offer a foundation for potential exploration of deeper qualitative questions in future research endeavors.

Table 16: Main Reasons Why High School Students Haven't Learned Programming

What's the main reason you haven't tried programming yet?	Department		Total
	Information Science ($n = 53$)	Electrical Engineering ($n = 38$)	($n = 91$)
No chance to learn	12	12	24
Not taught in school	11	3	14
No interest to learn	10	1	11
Looks difficult	3	2	5
Don't know where to start	2	3	5
No computer	0	5	5
No time	1	3	4
My environment	0	3	3
Can't learn by myself	3	0	3

5. Discussion and Conclusion

The outcomes of this questionnaire offer insights into the programming knowledge, computer access, interest, and usage among Japanese high school graduates. It was discovered that out of the 173 participants, 82 (47.4%) had prior exposure to programming before entering university. However, it is noteworthy that all 173 participants will be studying programming for a year during their first year of university and will undergo programs that equip them for careers often demanding programming skills. These statistics raise the question of why over half of the participants chose their majors without first-hand experience with the programming requirements in their future careers.

This question gains further importance when considering the findings in Sections 4.2 and 4.4 of this study. In section 4.2, it was revealed that among 20 students from the Information Science Department, the collective computer interest score was 4.89, below the 5.5 neutral score. Additionally, 11 of the 91 students without programming experience expressed no interest in learning programming. These findings suggest a significant number of students lack interest in computers and programming, despite selecting majors that necessitate programming skills and prepare them for careers in the technology industry. Despite a portion of participants displaying low interest in these areas, most students exhibited a strong interest in computers. As indicated in Table 7 in Section 4.2, most students had high computer interest scores, surpassing 6.8.

Surprisingly, despite students' high interest in computers and programming, most reported using a computer only a few times a week or even less frequently. One would expect individuals with a strong interest in these subjects to use computers more frequently. However, out of the 173 students, 101 reported using a computer most frequently at school. Furthermore, 113 students reported not having their own computer, and 76 students mentioned either a lack of computers at home or having only one computer shared among family members. These findings suggest the presence of socioeconomic or familial factors that limit high school students' access to computers. It raises questions such as whether computers are financially affordable for Japanese families, is computer literacy undervalued, or whether smartphones are viewed as sufficient replacements for computers in Japanese society. These are inquiries that the authors hope to investigate further in future research.

One notable finding of this study was that only approximately 20% of students with programming experience reported self-studying programming. Consequently, it can be concluded that around 80% of students acquired their programming knowledge solely through formal education. This observation, combined with responses indicating reasons for not having learned programming yet, such as "It's not taught in school" and "I can't learn by myself", suggests the existence of an underlying belief that programming is a subject that necessitates external guidance. Exploring the formation and underlying reasoning behind this belief is crucial since programming is a discipline that demands hands-on learning rather than passive observation. The perception that programming cannot be effectively learned independently or that it must be exclusively taught in a classroom setting raises concerns. If there is one curial lesson to impart in junior high school or high school programming lessons, it should be the recognition that technical skills can and should be pursued autonomously.

The final area of focus was on the actual programming proficiency of students. Of the 173 students, only 82 reported having learned, studied, or attempted programming. However, when examining specific languages, it was found that only 43% of the students with C experience and 69% of those with Python experience were able to correctly identify the respective “Hello World” programs in those languages. This discrepancy raises intriguing questions about how students can claim to have experience with a programming language while struggling to identify such a fundamental program in that language.

In summary, this study delved into the programming knowledge and various facets of computer usage and interest among 173 recent high school graduates. It successfully identified the programming languages students are learning, assessed the sources of their programming experience, and highlighted barriers hindering their programming education despite their expressed interest. Moreover, the study brought to the forefront concerns regarding students’ access to computers, motivations for enrolling computer-intensive majors despite limited interest, and the disconnect between self-reported language experience and the ability to recognize programs in that language.

6. Limitations and Further Research

Although participants were drawn from various regions and represented a diverse array of high schools, it is important to note that all participants were enrolled in the same university. Therefore, the findings may not fully capture the knowledge and abilities of Japanese high school graduates. Moreover, this study focused on students from two specific majors, which further limits the generalizability of the results.

This study acknowledges several limitations associated with the questionnaire used to assess programming knowledge. While the questionnaire provided insights into participants’ self-reported programming knowledge, a more practical approach would involve task-based assessments to evaluate actual programming skills. Furthermore, the use of only Python-specific questions may not fully align with participants’ programming experiences. Future studies could consider providing participants with the option to choose the programming language they are most comfortable with for answering questions.

The results of this study have raised numerous questions that warrant further investigation in future research. For instance, exploring the socio-economic factors associated with limited computer access and perceived lack of opportunity to learn programming would provide a better understanding of the barriers faced by certain students. Additionally, investigating how family values regarding technology influence computer access and preparedness for job opportunities would be valuable.

7. Acknowledgement

This work was partly supported by the Grant-in-Aid for Scientific Research (C) Grant Numbers 24501221, 16K01138, and 22K2926 from the Japan Society for the Promotion of Science as well as Fukui University of Technology Cluster Research Grant (Cluster D).

8. References

(1) Government of Japan. (2013). *Annual Economic and Fiscal Report*. https://www5.cao.go.jp/j-j/wp/wp-je13/index_pdf.html

(2) Kanemune, S. (2019). Programming Education and Tools for K12 Schools in Japan. *ICICE Communications Society Magazine, 13*(2), 92-99. https://doi.org/10.15.87/bplus.13.92

(3) Kanemune, S., Shirai, S, & Tani, S. (2017). *Olympiads in Informatics, 11*, 143-150. doi: 10.15388/ioi.2017.11.

(4) Nakamura, Y. (2021). *Programming Education in Japanese Schools Identification of Existing Barriers and Suggestions for the Teacher’s Online Platform* (Research Report – 0003) UCA-INSPE Academie de Nice. Retried from: https://hal.archives-ouvertes.fr/hal-03410836/document.

(5) Nishikawa, K., Misawa, R., & Takahashi, N. (2021). The bright and dark sides of programming education in Japanese elementary schools: Clarification of practical issues. *Bulletin of Center for Teacher Education and Development, Okayama University, 11*, 59-73. https://doi.org/10.18926/CTED/61565

(6) Vegas, E & Fowler, B. (2020). What do we know about the expansion of K-12 computer science education? *Brookings*. Retrieved from: https://www.brookings.edu/research/what-do-we-know-about-the-expansion-of-k-12-computer-science-education/

（2023年８月３日受理）