International Journal of Didactical Studies
Volume 6, Issue 1, 2025
https://doi.org/10.33902/ijods.202520935

Research Article

Mathematical thinking in programming education: Applied to the solution of linear equations in Junior high school

Takuma Takayama

Yamato University, Osaka, Japan (ORCID: 0009-0009-4125-8490)

The purpose of this paper is to clarify the significance of integrating programming into mathematics education. To do so, the study sought to propose and practice an instructional method for fostering computational thinking in mathematics education. Computational thinking is a thought process for formulating a solution to a problem so that it can be expressed as a computational procedure or algorithm. Mathematical thinking is closely related to computational procedures and algorithms. To confirm this, we conducted a practice of incorporating programming into a lesson on simultaneous equations for third-year junior high school students. The results suggested the following three points. (1) The usefulness of the Cramer's formula algorithm for simultaneous equations was understood. (2) By considering the bifurcation structure due to the existence of solutions to simultaneous equations, the understanding of the characterization of the existence of solutions by the determinant was deepened. (3) The students realized the usefulness of programming by utilizing programming to solve problems of simultaneous equations in the real world where the coefficients are complex, such as finding the ratio of copper and zinc alloys in brass. A post-class survey suggested that incorporating programming into mathematics education deepens students' understanding of mathematics.

Keywords: Computational thinking, Mathematical thinking, Programming

Article History: Submitted 31 March 2023; Revised 5 June 2024; Published online 17 August 2024

1. Introduction

The purpose of this study is to examine the significance of implementing programming education in junior high school mathematics departments, which has been pointed out as a necessity in recent years, through practice, and to consider the programming thinking that is fostered in such education. In the new Courses of Study, programming education is positioned as a cross-curricular, cross-curricular program, and its goal is to foster a programming mindset. The author hypothesized that incorporating programming education into mathematics education would not only foster programming thinking, but also enable students to think about mathematics more clearly.

The significance of incorporating programming in mathematics education is threefold.

(1) Mathematical equation processing is a strong point of programming, so it is possible to incorporate many problems and concepts that lead to machine learning.

(2) Programming and solving mathematical problems lead to the discovery of new mathematical properties and a better understanding of mathematics.

(3) Mathematical thinking is cultivated in the process of verifying programming.

In this study, we created a program to find solutions to simultaneous equations in the study of simultaneous equations in the second grade, and through this practice, we confirmed and verified the process of considering the understanding of Cramer's formula for simultaneous equations. As a result, it was

Address of Author

Takuma Takayama, Yamato University, 2-5-1, Katayama cho, Suita City, Osaka 564-0082, Japan.

✉ mtakayama.takuma@yamato-u.ac.jp

How to cite: Takayama, T. (2025). Mathematical thinking in programming education: Applied to the solution of linear equations in Junior high school. *International Journal of Didactical Studies, 6*(1), 20935. https://doi.org/10.33902/ijods.202520935

found that incorporating programming education into the class deepened the students' view of mathematics, and that programming thinking, mathematical thinking, and logical thinking (the ability to think logically) were required in the process.

2. Background

Teaching methods which incorporate program improvements within the class that are easy and efficient are a suitable teaching method of programing in mathematics class. Computational thinking and mathematical thinking play a very important role in the process of program improvement. According to Wing (2006), Computational Thinking is the thought process involved in formulating a problem and expressing its solution by means of computer. However, it is well known that we can do better programming by using mathematical thinking. Therefore, this study examines a scene where mathematical thinking is used in addition to computational thinking and proposes a new teaching method for programming in mathematics classes. Figure 1 indicates the relationship between computational thinking and mathematical thinking in this class and Figure 2 shows mathematical thinking and computational thinking work in classes that incorporate program improvements.

Figure1

The relationship between computational thinking and mathematical thinking in this class

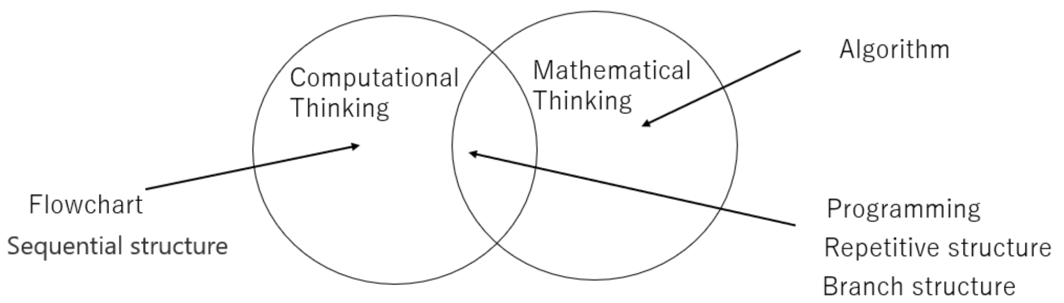
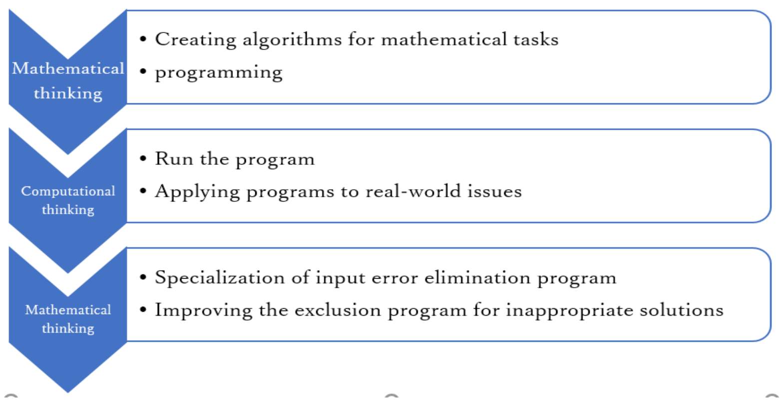


Figure 2

Computational thinking and mathematical thinking in this practice



Until now, researchers have mainly practiced four programming sectors of education:

- Simultaneous equations programs (Sequential structure, Branch structure)
- Binary to decimal conversion programs (Sequential structure)
- Decimal to binary conversion programs (Sequential structure, Branch structure)
- AD to zodiac conversion programs. (Sequential structure, Branch structure, Repetitive structure)

Each of these four programs include the steps in Table 1.

Table 1
Program steps in class practice

5 Programming steps involved above 4 lesson programs	*Input*	*Sequential structure*	*Branch structure*	*Repetitive structure*	*Output,Application to real problems*
Simultaneous equations	✓	✓	✓		✓
Binary to decimal	✓	✓			✓
Decimal to binary	✓	✓	✓	✓	✓
AD to zodiac	✓	✓	✓	✓	✓

In this study, we will examine the teaching method of programming from the practice of programming simultaneous equations.

3. Practice Methods

In this study, we conducted the following class in which students at a public junior high school in Tokyo were asked to create programming for simultaneous equations and use it to solve realistic problems.

3.1. Reasons for using the basic program in this class

In this class, we used the Basic Program. Basic Programs are languages that cannot use recursive calls (calling function A to function A). Therefore, it can be said that programming using mathematical thinking has significance in assembling the functions to be used from scratch. In addition, there are advantages that Basic Programs can be learned to some extent in a short time, the program flow can be easily grasped visually, and the program can be improved.

3.2. Aims of This Class (Simultaneous equations)

By associating previous mathematics with the contents of the program, the student performs programming including input and sequential structure, and learns programming including branch structure in the improvement of the program. In programming, we aim to create better programs by using computational thinking and mathematical thinking.

3.3. Participants

The following schedule was used to teach the following students:

・Target Students

28 second-year junior high school students from Tokyo public schools.

・Date and time:

First session: Monday, July 11, 2022
Second session: Wednesday, July 13, 2022

3.4. Flow and Goals of This Class

- Create a program for solving simultaneous equations with Cramer's Rule.
- Use a simultaneous equations program to solve real problems with numeric values that cannot be solved by the addition/subtraction method or substitution method easily
- To exclude the input errors by using branch structure of the program, in hopes that the results will be examined and further improved.

4. Study Context

In this class, students will create a program that calculates the solution to a simultaneous equation by entering the coefficients of the equation. The usefulness of the program will be verified by examining cases of real-world problems that are not covered in the textbook and that cannot be calculated by hand.

4.1. Programming to Solve for Solutions to Simultaneous Equations

The quadratic equation has a solution formula, but the simultaneous equation is calculated by the addition/subtraction method and substitution method. Therefore, the student does not learn the formula for the solution, but students realize that there is a formula for Cramer's Rule and other simultaneous equations in the case of $ad - bc \neq 0$. In order to create a program for finding solutions to simultaneous equations, students are allowed to notice that the value they enter is not a mathematical formula, but six coefficients.

$$\begin{cases} ax+by=e \cdots ① \\ cx+dy=f \cdots ② \end{cases} \Rightarrow \begin{cases} x=(ed-fb)/(ad-cb) \cdots ③ \\ y=(af-ce)/(ad-cb) \cdots ④ \end{cases} \Rightarrow$$

```
10 inputv "a=",a
20 inputv "b=",b
30 inputv "c=",c
40 inputv "d=",d
50 inputv "p=",p
60 inputv "q=",q
70 x=(p*d-b*q)/(a*d-b*c)
80 y=(a*q-p*c)/(a*d-b*c)
90 print "x=",x
100 print "y=",y
```

Given Program I is relatively simple with only a sequential structure, this was an opportunity for students to learn the programming language and its structure by creating this program. By using a program to solve the complex simultaneous equations of coefficients, an opportunity was given to consider its convenience.

4.2. Solving a Real-world Problem by Utilizing the Program (Solving problems related to brass synthesis)

Brass is an alloy of copper and zinc and has long been used to make Buddha statues in Japan. Currently, industry standards of Brass are established according to the ratio of copper and zinc. This time, programming was performed to determine the proportion of synthesis when making an alloy from copper and zinc.

4.3. Utilization of Simultaneous Equations

There are two reasons for treating the brass alloy problem with programming: this problem can be solved by simultaneous equations, and the coefficients of the simultaneous equations are too complicated to solve by hand.

Table 2
Real-world issues in this paper

C2600 copper 71% zinc 29%
C2801 Copper 61% Zinc 39%
I want to make 4.5kg of C2680, 66% copper.
Let's calculate how much each should be mixed.

The learner solves the brass problem in Table 2 by creating a simultaneous equation as follows

$a = 1, b = 1, c = 0.71,$
$d = 0.61, p = 4.5, q = 3.0825$

$$\Rightarrow \begin{cases} x+y=4.5 \\ 0.71x+0.61y=3.0825 \end{cases} \Rightarrow \begin{matrix} x=2.8125 \\ y=1.6875 \end{matrix}$$

4.4. Program Improvements (Programs including sequential structure and branch structure)

Using Program, I, which simply solves simultaneous equations, students were able to find a solution. Next, students notice that they must make the number of the human errors smaller at the input. Thus, after mathematical activity they come to realize they must eliminate the following three points:

- Unnecessary input error value. In other words, in this problem, a and b are always 1, and p remains a constant. As a result, these can be omitted by improving the program.

I want to make brass with a kg of brass with a copper content of q% by mixing brass with a copper content of a% and b%.

$$\Rightarrow \begin{cases} x+y=p \\ ax+by=q \end{cases}$$

- Consideration of ways to exclude obvious input mistakes. Here, students consider a program that returns to the beginning if a negative value is entered.

A major problem in entering data is typing errors. In this case, it is an opportunity to think about the branch structure of the program by excluding obvious input mistakes.

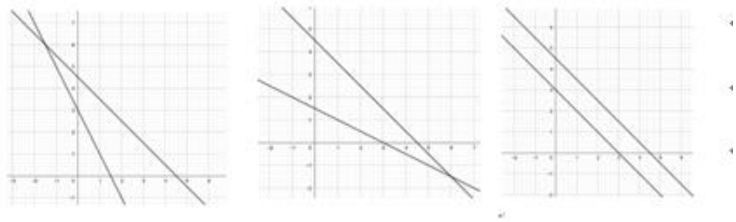
- Examination of solutions and consideration of ways to exclude values of solutions that do not fit for real world by using mathematical thinking.

This consideration allowed students to improve the coefficients of Program II. In examining the solution, if the coefficient is negative, it has already been excluded and it is sufficient to examine the solution under that assumption. Therefore, this simultaneous equation is considered in the graph as follows. Students discuss in groups of four and consider three cases to be excluded by the mathematical thinking. As a result, 71.4% of students were able to create Program III and a post-class survey revealed that almost all students were able to understand Program III.

In the brass composite problem, all coefficients in the simultaneous equations are positive. We have already improved the program to exclude incorrectly entered negative values. Therefore, the following mathematical graph considerations are necessary to exclude the calculated error values. It can be seen that there are three cases in which the calculated result is inappropriate That is, (1) the case where the value of x is negative, (2) the case where the value of y is negative, and (3) the case where the two graphs are parallel and have no solution. The learner should consider this in detail as shown in Figure 3.

Figure 3

Excluding by mathematical thinking



(1) case x <0 (2) case y <0 (3) case ad-bc=0

Figure 4

Process of program improvement

```
hajime:
INPUT "c=",c
INPUT "d=",d
INPUT "p=",p
INPUT "q=",q
x=(p*d-1*q*p)/(1*d-1*c)
y=(1*p*q-p*c)/(1*d-1*c)
PRINT "x=",x
PRINT "y=",y
```

Program II

```
Hajime:
INPUT "c=",c
INPUT "d=",d
INPUT "p=",p
INPUT "q=",q
IF c<=0 THEN
GOTO hajime
ELSE,
IF d <=0 THEN
GOTO hajime
ELSE,
IF p<=0 THEN
GOTO hajime
ELSE,
IF q<=0 THEN
GOTO hajime
ELSE,
x=(p*d-1* p*q)/(1*d-1*c)
y=(1* p*q-p*1)/(1*d-1*c)
IF x<=0 THEN
GOTO hajime
ELSE,
IF y<=0 THEN
GOTO hajime
ELSE,
IF d-c=0 THEN
GOTO hajime
ELSE,
PRINT "x=",x
PRINT "y=",y
```

Program III

5. Outline of Practice

The class was divided into two 50-minute lessons: the first hour was devoted to deriving Cramer's formula and reviewing basic programming issues; the second hour was devoted to programming simultaneous equations and their application to real-world problems.

5.1. First Session: Monday, July 11, 2022

The first hour of the two-hour class is as follows.

Contents: Programming of linear equations

- Confirmation of solution formulas for linear equations
- Introduction to programming for linear equations
- Confirmation of input and output parameters for linear equations
- Exclusion program when 0 is input for coefficients
- Explanation of exclusion program when a≠0
- Derivation of Cramer's formula

5.2. Second Session: Friday, July 15, 2022

The second hour of the two-hour class is as follows.

Contents: Programming of simultaneous equations

- Programming of quadratic equations using Cramer's formula
- Solve problems other than ad-bd≠0 with the program and confirm that the program is correct.
- Check the case of ad-bd=0, and think about the meaning of the error.
- Display two graphs by GeoGebra and confirm that they are parallel.
- Solve the brass alloy problem with the program.

5.3. Practice Details

The following is a detailed description of the class, including protocols. In addition, Figures 5 through 14 are screens that were sent and shared using educational support software to the students' tablets in class.

5.3.1. Students' learning up to the previous class

Following the instructional plan in Section 4, we taught programming of simultaneous equations to second-year junior high school students in Tokyo. They have already finished learning up to solving simultaneous equations. However, most students have never learned about programming. In addition, the relationship between simultaneous equations and linear expressions and the use of simultaneous equations for real-world problems has not yet been learned.

5.3.2. Programming linear equations

We show the content of the first hour of a two-hour class. At the beginning of the class, a brief overview of programming and programming languages were reviewed. After that, the programming of linear equations was confirmed in the following sequence. Confirmation of solution formulas for linear equations. The protocol of the class is shown below. The protocol of the class is S indicate as student and T as a teacher.

> T: What shape did the linear equation have?
> S: Consider the formula for the solution of ax+b=c. In other words, find the formula for the value of x from the three values of the linear equation a, b, and c.
> T: How can I do that?
> S: I think we should solve for x.
> T: Please solve it.
> S: x = (c-b)/a.
> T: That's right. With this formula, we can find the value of x from the three values a, b, and c.
> S: By making a program, we can input three values of a, b, and c and find the value of x.

Linear equations are studied in the first grade of junior high school, but no solution formulas are created. The simultaneous equations studied in the second grade of junior high school also do not have solution formulas. However, in quadratic equations studied in the third grade of junior high school, students learn how to solve equations by factorization, and then learn solution formulas. It is important to realize the necessity of solution formulas for linear equations in programming.

5.3.3. Introduction to linear equation programs

Due to time constraints, the program for linear equations was initially distributed to the students using the school tact, which was used to project the program in the BASIC language to the students at the same time. Then, the flow of input, processing, and output was confirmed. After that, we had the students open Webasic, an online basic programming site. Then, I asked the students to type in the program I was sending them. Some students made errors, but I made them understand that this experience is important in programming. For students who were not able to type well due to time constraints, we copied and pasted the program that we had sent to School Tact to enter the program. They then entered each parameter for 2x + 3 = 15 and verified that the program was working correctly.

5.3.4. Derivation of Cramer's formula.

The students were asked to focus on the scene of the processing of a linear function program. When creating a program for simultaneous equations, I asked students to answer which numbers to input, what processing to perform, and which numbers to output.

The students commented that they needed to input six numbers, that they needed a solution formula for simultaneous equations as well as a program for linear equations, and that they needed to output two numbers, x and y, as the answer.

In this way, it was confirmed that for the two equations ax+by=p and cx+dy=q, the six letters a, b, c, s, p, and q must be input. And student realize a necessity of formula for finding two solutions, x and y, from these six values and that x and y are output. Although Cramer's formula is not handled in junior high school, we were able to overcome the difficulty by transforming the formula using the add-subtract method of simultaneous equations and by stating the general case together. Initially, the coefficients of x were aligned by multiplying ① by c and ② by a. By taking the difference of the two equations, x was eliminated and the formula for y was derived. Next, in the same way, by aligning the coefficients of y and taking the difference of the two, the formula for x was derived by eliminating y (see Figures 5 and 6).

Figure 5
Derivation of Cramer's formula step 1

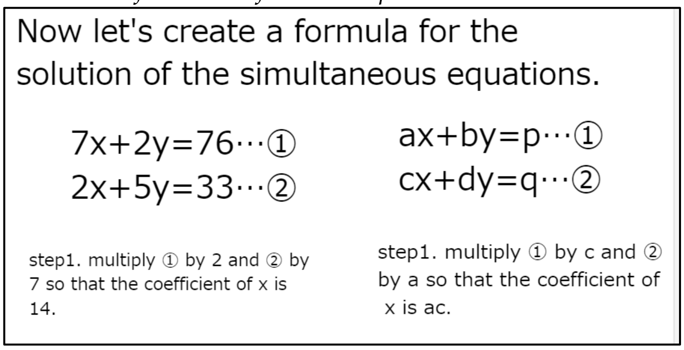
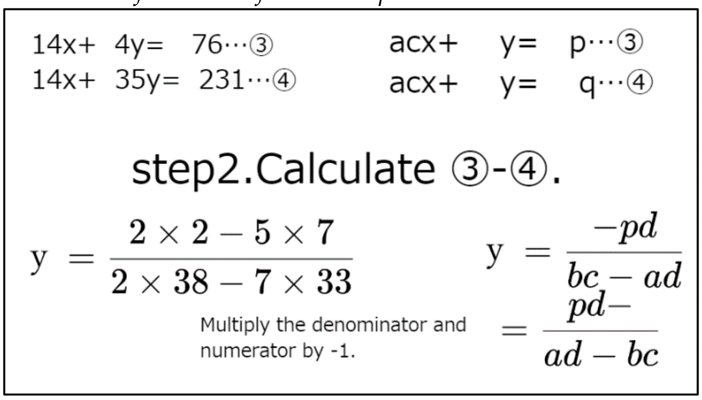


Figure 6
Derivation of Cramer's formula step 2



5.3.5. Programming of simultaneous equations

The following is the content of the second hour of the two-hour class, including the protocol. The goal of this lesson is to create a program to find solutions to simultaneous equations using Cramer’s formula created in the first hour. The goal is then to use that program to solve a real-world problem, as well as to improve upon that program.

5.3.6. Execution of the program

The class began with a review of the previous period. First, the program for linear equations was reviewed, followed by a review of the formulas for the solutions of simultaneous equations. After that, by improving the program for linear equations, the students created a program for deriving the solution of simultaneous equations.

The protocol of the scene of checking the input and output parameters of the simultaneous equations is shown below.

> T: Let's create a program to derive solutions to simultaneous equations in the same way as programming to derive solutions to linear equations.
> T: For a linear equation, we created a program to find the value of x by inputting three variables, a, b, and c, and using the solution formula. How many variables should be input for simultaneous equations?
> S: I think four.
> T: Now let's write the simultaneous equations in symbols.
> ax+by=p
> cx+dy=q
> S: I think we only need to input six variables: a, b, c, d, p, and q.
> T: What else do you need?
> S: I think it is the solution formula.

Most students were able to create a program to find the solution to the simultaneous equations. Two simultaneous equations were then solved using the program they had created: the first dealt with a textbook problem whose answer was an integer value; the second was the case where the determinant was zero, i.e., where there were an infinite number of solutions. The students who solved the first simultaneous equation with the program expressed a sense of relief. However, when I entered the second system of simultaneous equations, the students' hands went up one after another. I explained to the students that the reason for the error was that when the coefficients of the x's were aligned when calculating by addition and subtraction, the coefficients of the y's were also aligned, and that the denominator of the program was zero. I confirmed that the cause of the error is that the coefficients of y are also aligned when the coefficients of x are aligned when calculating by the add/sub method and the denominator of the program is 0 (Figure 7).

Figure 7
Trial of a linear equation program

Solve the following equations using the program

(1) $3x + 6 = 12$

(2) $0x - 5 = 1$

5.3.7. Application of the program to real-world problems

The activity utilized the program to solve the problem of brass in Buddhist statues. The students asked questions about the problem, which showed that the students were interested in the project. The flow of the class was as follows (Figure 8).

Figure 8
Formulation of a real-world problem

Buddha images are made of brass. Brass is an alloy of copper and zinc.
We have two alloys on hand: C2600, which is 69% copper and 31% zinc. The C2801 alloy is 61% copper and 39% zinc.
How many kilograms of each of C2600 and C2801 should be melted to make a 4.5 kilogram Buddha statue of alloy C2680, which is 64% copper and 34% zinc?

①What is x and y?

②Let's make a simultaneous equation II.

```
c=0.69
d=0.61
p=4.5
q=0.64
x=  Let's make a simultaneous
equation II.
1.6874999999999982
y=
2.812499999999997
```

The students then have a total weight of p kg of copper x kg and zinc y kg, so the coefficients of the equation in ① are always x + y = p. Therefore, the students realized that for ①, only the value of p needs to be entered (see Figure 9).

Figure 9
Considerations for real-world problem

I want to mix xkg of brass A with c% copper and ykg of brass B with d% copper to make brass Pkg with q% copper. How many kg of each should I mix?

$$x + y = p \ldots ①$$
$$cx + dy = pq \ldots ②$$

Let's improve Program I into a program that finds the solution to this simultaneous equation.

5.3.8. Program improvements

I had them come up with a program that would return to the beginning as an error if the number entered was negative. To get them to think about that problem, I gave them the following linear equation as an example (Figure 10). The following simultaneous equations were given as examples (Figure 10). Then, the program for the simultaneous equations was refined based on this (see Figure 11).

Figure 10
Example of exclusion program when 0 is entered as a coefficient in a linear equation

If a=0, an error occurs, in which case the program should be modified to go back to the beginning.

```
10 input "a=",a
15 if a=0 then goto 10
20 input "b=",b
30 input "c=",c
40 x=(c-b)/a
50 print "x=",x
60 end
```

Figure 11

Example of an exclusion program when a negative number is entered for the coefficients of a simultaneous equation

Improve the program to exclude input errors.

```
30 input "c=",c
40 input "d=",d
50 input "p=",p
60 input "q=",q
70 x=(p*d-1*p*q)/(1*d-1*c)
80 y=(1*p*q-p*c)/(1*d-1*c)
90 print "x=",x
95 print "y=",y
100 end
```

```
30 input "c=",c
35 if c<0 then goto 30
40 input "d=",d
35 if d<0 then goto 30
50 input "p=",p
35 if p<0 then goto 30
60 input "q=",q
35 if q<0 then goto 30
70 x=(p*d-1*p*q)/(1*d-1*c)
80 y=(1*p*q-p*c)/(1*d-1*c)
90 print "x=",x
95 print "y=",y
100 end
```

When all the coefficients to be input for the simultaneous equations are positive values, there are only three cases where the graphs of the two unary linear equations are positioned as shown in Figure 9. Therefore, since the three negative values are excluded as errors, the remaining errors are the three cases, ① when the x value of the solution of the simultaneous equations is negative, ②when the y value is negative, and ③ when the two lines are parallel and no solution exists (see Figure 12). Then, this program was improved (see Figure 13).

Figure 12

Considerations for solutions of simultaneous equations

In this case, consider the possible cases for the sign of the output result.

$$x + y = p$$
$$cx + dy = pq$$

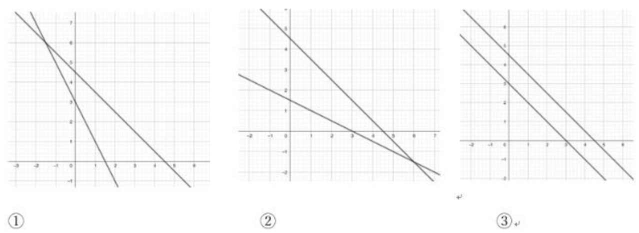


Figure 13

Exclusion program for negative solutions of simultaneous equations

Modify the program to return to the beginning if the result is negative.

```
30 input "c=",c
35 if c<0 then goto 30
40 input "d=",d
35 if d<0 then goto 30
50 input "p=",p
35 if p<0 then goto 30
60 input "q=",q
35 if q<0 then goto 30
70 x=(p*d-1*p*q)/(1*d-1*c)
80 y=(1*p*q-p*c)/(1*d-1*c)
90 print "x=",x
95 print "y=",y
100 end
```

```
30 input "c=",c
35 if c<0 then goto 30
40 input "d=",d
35 if d<0 then goto 30
36 if c=d then goto30
50 input "p=",p
35 if p<0 then goto 30
60 input "q=",q
35 if q<0 then goto 30
70 x=(p*d-1*p*q)/(1*d-1*c)
80 y=(1*p*q-p*c)/(1*d-1*c)
82 if x<0 then goto 30
83 if y<0 then goto 30
90 print "x=",x
95 print "y=",y
100 end
```

The students were able to rule out cases ① and ②, as shown in Figure 13. Therefore, they confirmed that the case when the slopes of the two lines coincide is when the determinant is zero (see Figure 14).

Figure 14

Consideration of parallel condition

$$ax+by=p \cdots ①$$
$$cx+dy=q \cdots ②$$

In general, let us try to find the conditions under which the slopes of the graphs ① and ②above coincide.

6. Data Analysis and Results of Pre- and Post-survey

A pre-survey was conducted before the class and a post-survey was conducted after the class. This survey was conducted on 29 people who attended two classes. The results are as follows.

6.1. Pre- and Post-survey Questions

In this practice, the same questions were used for the pre- and post-surveys. The following questions were asked to determine whether programming improves academic performance in mathematics. These questions were asked to determine whether the students were able to determine variables, formulate simultaneous equations, and find solutions to simultaneous equations (see Table 3).

Table 3

Real-world problems that were submitted to the pre-survey and post-survey

(1) We are considering room assignments for 50 third-year students who will be staying at the convention. There are two types of accommodations: a 4-person type for 3,500 yen and a 3-person type for 3,000 yen. How many rooms should be allocated for each to bring the total accommodation cost to 46,000 yen?

①Let x denote the number of ( ) and y denote the number of ( ).

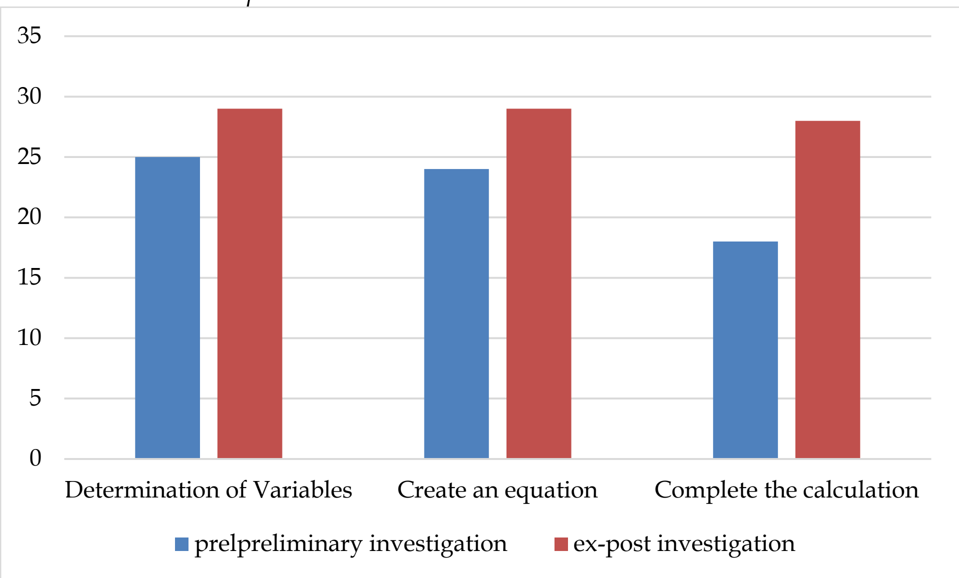
②Create a simultaneous equation.

③Find the answer to the question.

The results show that creating programming for simultaneous equations increased the students' academic skills in all of the following areas: letter selection, equations, and calculations in written problems of simultaneous equations (see Figure 15).

Figure 15

Answers to real-world problem



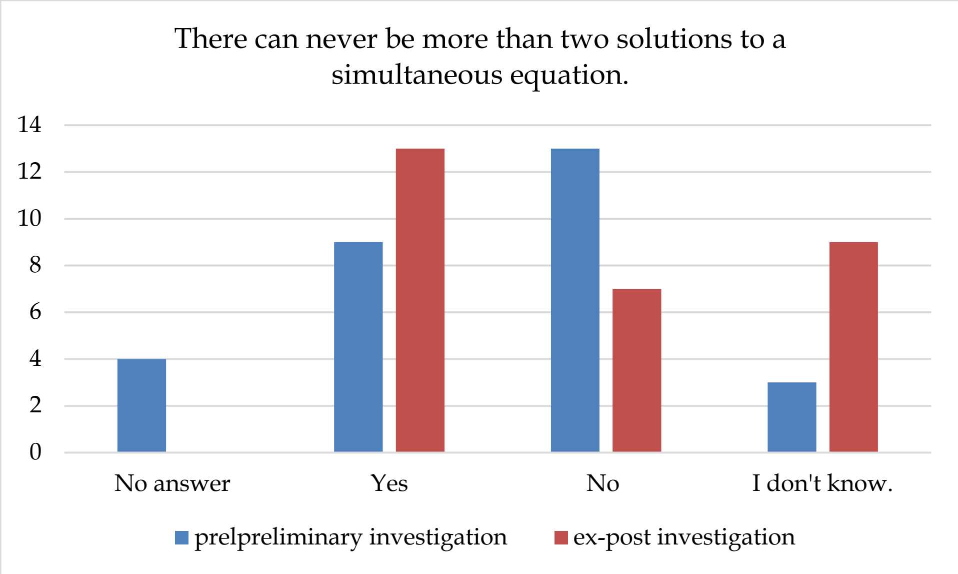
6.2. Pre-and Post-survey Results

Questions 1 through 5 examined students' awareness of how much their understanding of simultaneous equations was deepened by creating a program to find solutions to simultaneous equations, using a questionnaire. In addition, the students' attitudes toward programming were investigated in a post-class questionnaire to determine how their attitudes toward programming changed after the class.

6.2.1. Question 1: "There can never be more than one solution to a simultaneous equation."

It can be seen that the understanding of the fact that no more than three solutions of simultaneous equations exist is also deepened in this class (see Figure 16).

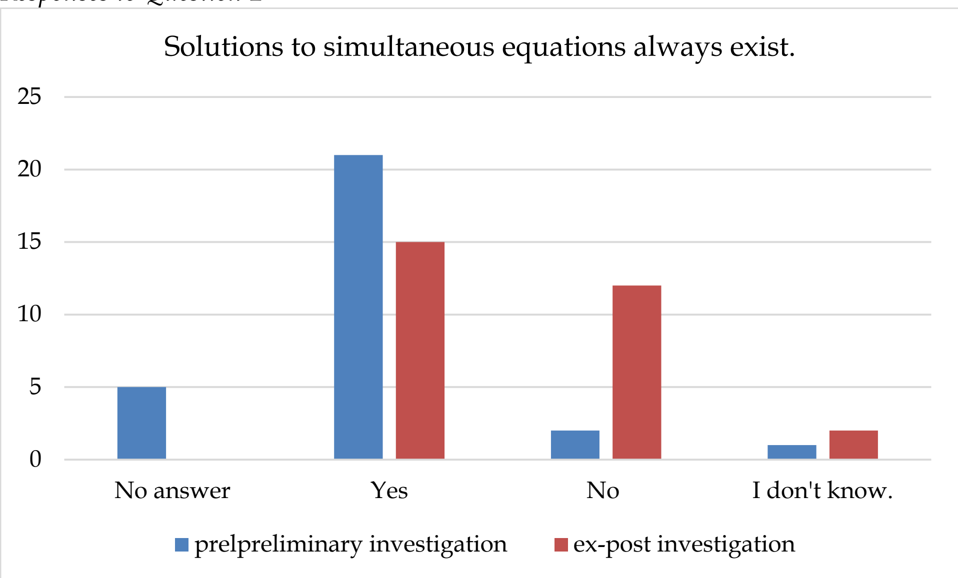
Figure 16
Responses to Question 1



6.2.2. Question 2: "There is always a solution to a simultaneous equation."

Programming to find solutions to simultaneous equations has deepened the understanding that solutions to simultaneous equations do not always exist (see Figure 17).

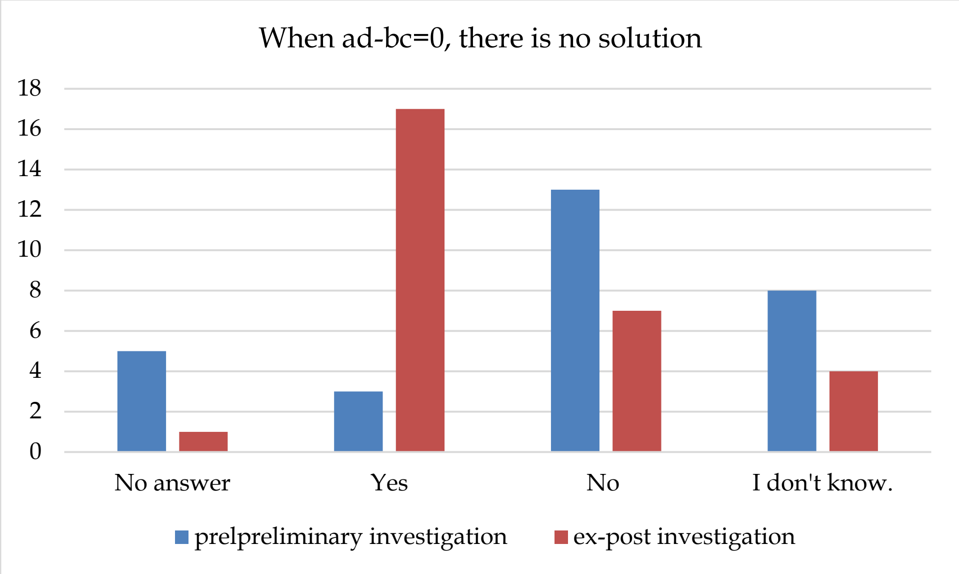
Figure 17
Responses to Question 2



6.2.3. Question 3: "In a simultaneous equation, if ad-bc=0, then no solution exists."

In this lesson, the students did not consider the case where two straight lines coincide when ad-bc=0. Therefore, students thought that when ad-bc=0, the two lines were parallel and no solution existed (see Figure 18).

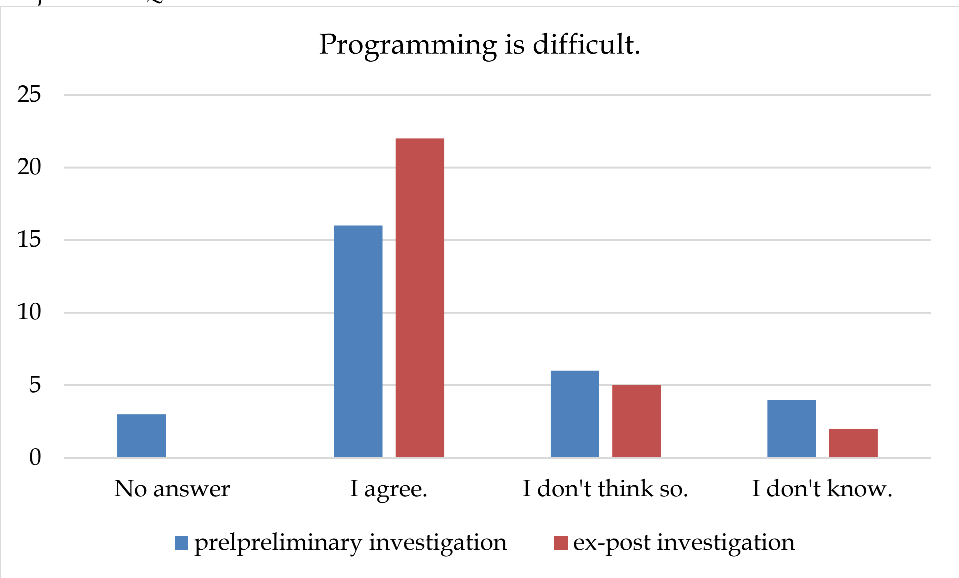
Figure 18
Responses to Question 3



6.2.4. Question 4: "Programming is difficult."

Most of the students had not learned programming before. Therefore, the students found programming difficult due to the lack of time relative to the content, as we found out in the post-class interview (Figure 19).

Figure 19
Responses to Question 4

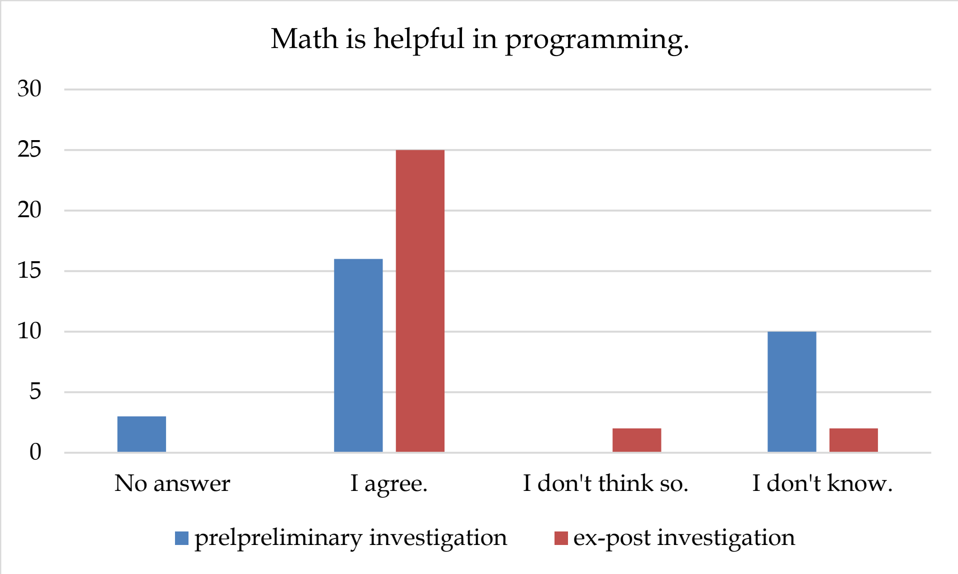


6.2.5. Question 5: "Mathematics is useful for programming."

The usefulness of mathematics in programming was perceived by most students (see Figure 20). Especially from the post-class feedback, students realized that solution formulas are very important in programming.

They also realized that even if the numbers are complex in real-life problems, they can all be solved once the programming is created.

Figure 20
Responses to Question 5



7. Discussion

The purpose of this study is to examine the significance of mathematics learning activities incorporating programming by examining the results of actual lessons. The results of the class practice in this study showed that programming for simultaneous equations deepens students' understanding of simultaneous equations. Statistical results and students' classroom activities suggested that programming for solving simultaneous equations improves students' mathematical academic skills, including the ability to determine the variables of simultaneous equations, to construct equations, to find solutions to equations, and to relate the relationship between the number of intersections of a graph and the slope of two straight lines.

Students also noticed two things in the process of learning this lesson. The first is the usefulness of the solution formulas for simultaneous equations. The solution formulas for simultaneous equations are essential for programming, and the students reaffirmed their usefulness. The second is the effectiveness of programs to find solutions to simultaneous equations. In this lesson, as a subject for solving real-world problems with mathematics, the students were asked to find out in what proportions copper and zinc are mixed as components of brass, a material used to make Buddhist statues in Japan. The students realized that real-world problems are very difficult for them to solve computationally because the coefficients are not simple integers, and that programming using solution formulas is effective.

Furthermore, through this class, students were able to understand the basic structure of programming. Knuth (1981) stated that "I tend to think of algorithms as encompassing the whole range of concepts dealing with well-defined processes, including the structure of data that is being acted upon as well as the structure of the sequence of operations being performed". Learning programming to solve simultaneous equations is a thought process in which a problem is formulated and its solution is expressed as a computational procedure or algorithm. In this regard, Aho (2011) states that "finding or devising appropriate models of computation to formulate problems is a central and often nontrivial part of computational thinking".

Therefore, it is suggested that mathematics learning activities incorporating programming can lead to the development of what the Aho (2011) calls "computational thinking." In the future, it will be necessary to develop lessons that introduce programming for finding solutions to quadratic equations and for finding solutions to trigonometric, exponential, and logarithmic equations, and to clarify the computational thinking that is required in such programming.

8. Conclusions

From the observation of students in the class and the questionnaire results after class, the following was found:

- Teaching methods incorporating program improvements are easy and efficient, thus they are suitable as teaching methods for early programming classes in the junior high school.
- Mathematical thinking and computational thinking play an important role in the thinking of the program with branch input errors and excluding inappropriate solutions.
- Students can realize the usefulness of mathematics and computers by creating a program on mathematics learned utilizing real-world problems.

Acknowledgements: This paper is a revised version of a paper presented at ICME-14 (2021) The 14th International Congress on Mathematical Education.

Declaration of interest: The author has no conflict of interest to declare regarding the content of this article.

Funding: No funding source is reported for this study.

References

Aho, A. V. (2011). *What is computation?* Association for Computing Machinery. https://doi.org/10.1145/1922681.1922682

Knuth, D. H. (1981). Algorithms in modern mathematics and computer science. *Springer-Verlag Lecture Notes in Computer Science*, *122*, 82–99. https://doi.org/10.1007/3-540-11157-3_26

Kong, S.-C., & Abelson, H. (2019). *Computational thinking education*. Springer Open. https://doi.org/10.1007/978-981-13-6528-7

Martin, S., & Jacobsen, M. (2018). Coding and computational thinking in math and science. *Alberta Science Education Journal*, *45*(2), 17–27.

Neira, R. H., Garcia-Iruela, M., Connolly, C. (2021). Developing and assessing computational thinking in secondary education using a TPACK guided scratch visual execution environment. *International Journal of Computer Science Education in Schools*, *4*(4). https://doi.org/10.21585/ijcses.v4i4.98

Perez, A. (2018). A framework for computational thinking dispositions in mathematics education. *Journal for Research in Mathematics Education*, *49*(4), 424–461. https://doi.org/10.5951/jresematheduc.49.4.0424

Wing, J. M. (2006). Computational thinking. *Journal for Research in Mathematics Education*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215