Journal on Mathematics Education
Volume 15, No. 2, 2024, pp. 613-638

Cross-cultural insights on computational thinking in geometry: Indonesian and Japanese students' perspectives

Rully Charitas Indra Prahmana[1,2,*], Satoshi Kusaka[3], Nur Robiah Nofikusumawati Peni[1,2], Hiroyuki Endo[3], Ahmad Azhari[4], Kanako Tanikawa[3]

[1]Mathematics Education Department, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[2]Ethno-Realistic Mathematics Education Research Center, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[3]Human Education Department, Naruto University of Education, Tokushima, Japan
[4]Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
*Correspondence: rully.indra@mpmat.uad.ac.id

Received: 3 January 2024 | Revised: 25 February 2024 | Accepted: 15 March 2024 | Published Online: 3 April 2024
© The Author(s) 2024

Abstract

Current research indicates the presence of highly skilled and motivated students with robust computational thinking backgrounds seeking opportunities to leverage their expertise in driving innovation and success in this era. These studies also reveal that students' computational thinking skills vary widely depending on educational resources, curriculum emphasis, and individual aptitude. Nonetheless, there is a growing recognition of the importance of fostering these skills, with efforts underway to integrate them more comprehensively into education systems worldwide, including in Indonesia and Japan, as representatives of developing and developed countries. Therefore, assessing the competency of computational thinking in these two countries would be intriguing. The descriptive qualitative research method was employed to delineate the computational thinking competencies of students in Indonesia and Japan. Student worksheets, specifically designed for this purpose, were utilized to gauge the development of these competencies during the learning process using the Scratch application. The results revealed that students employed various strategies in solving the given geometry problems. On the other hand, geometry is one of the mathematics topics that can identify students' computational thinking using this application. These findings were utilized to categorize students' computational thinking skills in the two countries and to identify potential obstacles students experienced in their efforts to enhance these skills. Nevertheless, these constraints offer significant insights into potential areas for future investigation and enhancement. Subsequent endeavors could prioritize conducting experiments by implementing specific learning approaches or methods that have demonstrated effectiveness in improving students' computational thinking skills. This study not only underscores the potential for expanding research on students' computational thinking skills but also provides an overview of the learning process, learning culture, and students' competence in solving geometry problems with tiered difficulty levels using their computational thinking skills.

Keywords: Computational Thinking Skills, Geometry, Indonesia and Japan, Primary School, Scratch

How to Cite: Prahmana, R. C. I., Kusaka, S., Peni, N. R. N., Endo, H., Azhari, A., & Tanikawa, K. (2024). Cross-cultural insights on computational thinking in geometry: Indonesian and Japanese students' perspectives. *Journal on Mathematics Education, 15*(2), 613-638. http://doi.org/10.22342/jme.v15i2.pp613-638

Prior research underscores the significance of computational thinking skills for students across diverse domains, enhancing problem-solving, collaboration, and analytical abilities, and can be developed through specific educational tools and approaches (Yadav et al., 2017; Ardito et al., 2020; Saritepeci, 2020; Molina-Ayuso et al., 2022; Israel-Fishelson & Hershkovitz, 2022; Yunianto et al., 2023). Yadav et al. (2017) advocate for integrating computational thinking as a crucial skill in the twenty-first century, emphasizing its inclusion in primary and secondary education to augment students' analytical capabilities. Computational thinking is essential, rooted in problem-solving and logical reasoning and drawing upon principles from computer science and mathematics (Ardito et al., 2020). These skills are valuable not only for students interested in pursuing careers in technology but also for all students in various disciplines (Molina-Ayuso et al., 2022). They prepare students for careers in technology and foster critical thinking, creativity, and problem-solving abilities that are valuable across multiple fields and throughout life (Saritepeci, 2020; Israel-Fishelson & Hershkovitz, 2022). Consequently, developing computational thinking skills is indispensable for all students, irrespective of their future professional endeavors.

Scratch embodies a visual programming paradigm meticulously designed for educational endeavors, specifically tailored to impart coding concepts to novices and juveniles (Stewart & Baek, 2023; Yunianto et al., 2023). It eliminates the need for learners to familiarize themselves with traditional programming languages such as Python and JavaScript. Distinguished for its accessibility, Scratch stands out as one of the most straightforward visual programming tools, empowering users to create projects spanning from games to educational materials. Engaging with Scratch allows students to dissect tasks into discrete components, employ loops for iterative processes, and utilize conditionals for informed decision-making, enabling educators to assess their aptitude for algorithmic reasoning (Fagerlund et al., 2020). At the core of Scratch's effectiveness lies its modular organization of code into reusable blocks, a design philosophy that encourages learners to distill recurring patterns and procedures, thus nurturing foundational aspects of computational thinking (Zhang & Nouri, 2019). Furthermore, Zhao et al. (2022) delve into the intricate nature of Scratch projects, often necessitating the resolution of multifaceted problems, compelling students to decompose overarching issues into more tractable constituents, thereby offering invaluable insights into their decomposition skills. As students immerse themselves in creating and debugging Scratch projects, they inevitably encounter coding patterns, learn to discern and harness them, and cultivate strategies for troubleshooting, thereby persistently overcoming obstacles.

Numerous studies have indicated that the utilization of Scratch applications in educational settings can markedly enhance and evaluate various dimensions of students' computational thinking skills across diverse academic levels and disciplines (Piedade & Dorotea, 2022; Molina-Ayuso et al., 2022; Gökçe & Yenmez, 2023). Piedade and Dorotea (2022) observed that participants in the experimental group who engaged in Scratch-based activities exhibited higher scores on the Beginners Computational Thinking Test than their counterparts who did not partake in such activities. Furthermore, Molina-Ayuso et al. (2022) documented notable enhancements in computational thinking skills among pre-service primary teachers undergoing mathematics education training through instructional modules featuring Scratch applications. Given that computational thinking encompasses problem-solving and logical reasoning abilities intrinsic to programming, activities based on Scratch serve as pragmatic instruments for skill cultivation (Gökçe & Yenmez, 2023). Conversely, Jiang and Li (2021) advocate for providing primary school students with more meaningful programming problems and integrating Scratch with subjects like mathematics and robotic programming to foster interdisciplinary learning and enhance computational thinking skills. Considering these findings, integrating Scratch applications into mathematics learning settings holds substantial promise for measuring and assessing students' computational thinking skills, particularly among primary school students.

Moreover, Scratch promotes creativity and innovation, enabling students to express their ideas through unique projects requiring creative thinking and computational concepts (Marcelino et al., 2018; Fagerlund et al., 2020). Evaluating the originality and complexity of their creations provides valuable insights into their creative problem-solving skills. Furthermore, Scratch projects often involve collaboration, encouraging students to work together, share ideas, and communicate effectively within project teams (Roque et al., 2016). Assessing the quality of students' collaboration and communication within Scratch project teams offers additional insights into their computational thinking skills. In summary, Scratch is a versatile platform for developing and accessing various facets of computational thinking, providing educators with valuable tools to support students' cognitive growth and skill development. However, it's essential to complement Scratch-based assessments with other evaluation forms to ensure a holistic understanding of students' skills and competencies.

Assessment via geometry problems can be seamlessly integrated into educational practices, employing Scratch to gauge students' prowess in computational thinking. Students can effectively use Scratch to solve geometry problems and enhance their computational thinking skills (Molina-Ayuso et al., 2023). Scratch facilitates visualization of geometric concepts by creating sprites representing shapes and properties, allowing students to experiment with various configurations and visually observe geometric relationships (Rodríguez-Martínez et al., 2020). Students develop algorithmic thinking and problem-solving abilities by breaking down problems into algorithms and designing scripts to manipulate geometric elements (Kale et al., 2018). The interactive interface of Scratch fosters real-time feedback, nurturing a profound comprehension of mathematical concepts intertwined with coding proficiency (Smith et al., 2022). Therefore, collaboration and sharing features on the Scratch platform promote peer learning and diverse problem-solving approaches, making it an engaging tool for students to apply computational thinking to geometry problem-solving tasks.

Issues about computational thinking skills exhibit variations between developing and developed nations yet share commonalities (Grover & Pea, 2013; Dahlman, 2007; Ausiku & Matthee, 2023). Developing countries like Indonesia need help accessing technology and the internet due to infrastructure deficiencies and economic obstacles (Sparrow et al., 2020). In contrast, a digital disparity persists within developed nations like Japan, disproportionately affecting marginalized demographics (Ono & Zavodny, 2007; Robinson et al., 2015). Both contexts confront hurdles in integrating computational thinking into educational frameworks owing to outdated curricula, insufficient teacher training, and resource inadequacies (Ogegbo & Ramnarain, 2022). Globally, computational thinking has pronounced disparities in specific regions. Awareness and perception regarding the significance of computational skills may be lacking, exacerbated by language barriers in non-English-speaking areas such as Indonesia and Japan. Mitigating these challenges necessitates investment in infrastructure, teacher development, curriculum innovation, and advocacy efforts to underscore the pivotal role of computational thinking for future proficiency in a digitized society.

Indonesia's educational system is increasingly prioritizing computational thinking skills to equip students with problem-solving abilities crucial for navigating the demands of the digital era, leveraging the curriculum's emphasis on problem-solving, mathematics, and creativity alongside initiatives integrating coding education and technology, all fostering an environment conducive to skill development and innovation (Global Education Monitoring Report Team - SEAMEO Regional Open Learning Center, 2023). On the other hand, recognizing the growing importance of computational thinking skills in Japan, where educational emphasis on problem-solving, mathematics, creativity, collaboration, and technology integration fosters an environment conducive to skill development, particularly evident through initiatives introducing coding education and leveraging the country's technological advancements in website-based learning environment to prepare students for success in the modern digital world (Kobayashi & Hasegawa, 2020; Kobayashi et al., 2022). Therefore, over the past decade, Indonesia and Japan have demonstrated a notable commitment to fostering computational thinking skills among students by developing conducive learning environments and comprehensive supporting tools.

This research constitutes a comprehensive effort to describe and classify one of the crucial skills students must possess in this era: computational thinking. Our team has developed a learning trajectory to support student worksheets organized into three sessions to accomplish this objective. The student worksheets contain activities to assess students' computational thinking skills based on their performance throughout the learning process. Through integrating learning activities and developed worksheets, this study presents a holistic approach to identifying and categorizing elementary school students' comprehension of fundamental geometric concepts and computational thinking skills. The selection of elementary school students stems from the belief that their capabilities can be maximized if cultivated from an early age.

Furthermore, the decision to implement the study in two countries, Indonesia and Japan, aims to investigate computational thinking capabilities across developing and developed nations. Consequently, the study poses two research questions: to what extent can the designed learning promote the computational thinking skills of elementary school students in Japan and Indonesia, and to what extent can the designed learning enhance students' understanding of Geometry based on their computational thinking skills in the countries as mentioned earlier. These research questions were explored to ascertain and classify students' computational thinking skills and understanding of basic geometric concepts. Through exploring these inquiries, we aim to gain insights into the efficacy of the developed learning tools in analyzing and comprehending the intricate facets of elementary school students' computational thinking abilities in Indonesia and Japan.

The following section of this article outlines the research methods utilized for data collection and analysis. Furthermore, the results and discussion section elaborate on the learning phases, which are divided into three sessions. Each session encompasses various activities, employing the student worksheet developed alongside geometry problems that were solved through Scratch. Lastly, the study findings demonstrate a significant improvement in students' computational thinking skills by implementing innovative student worksheet activities utilizing Scratch. Summaries of these results are presented in the concluding section.

METHODS

This descriptive qualitative inquiry endeavors to characterize innovative learning methodologies facilitated by Scratch, aimed at assessing students' computational thinking skills in the context of geometric problem-solving. The research methodology entails a sequential progression comprising three distinct phases: preparation, implementation, and analysis. During the preparatory phase, the research team formulated Scratch-based instructional materials, encompassing students' worksheets, programming tasks, and assessment tools, all of which were made accessible via an online platform (https://s.uad.id/Courses). Rigorous validation procedures were implemented, entailing evaluation by three lecturers from each respective side, two pre-service mathematics instructors, one scholar, and three subject matter experts, augmented by forum group discussions.

One of the principal advantages inherent in descriptive qualitative research lies in its steadfast emphasis on context, comprehensively integrating social, cultural, and environmental factors that influence phenomena under investigation. This perspective proves particularly invaluable in exploratory research endeavors, where the primary objective is to elucidate variables, formulate hypotheses, and establish a foundational comprehension for future inquiries. By prioritizing the observation of real-world experiences as they naturally unfold, qualitative research unveils the intricate and subtle dynamics inherent in human behavior and social relationships, aspects often overlooked by quantitative methodologies (Colorafi & Evans, 2016). The significance of studying phenomena within their authentic contexts is further underscored by Willis et al. (2016) in their comparative analysis of descriptive phenomenological and qualitative description research methodologies.

Subsequently, during the implementation phase, a standardized instructional model was adopted across educational settings in Indonesia and Japan, encompassing three instructional sessions as outlined in forthcoming intervention lessons sections. Data collection during the implementation stage involved direct observation, with the student-teacher researcher assuming the instructor role while other researchers in the project observed the entire class in both countries. Video and voice memos were utilized to record the lesson implementation to enhance subsequent analysis. These interventions, employing Scratch as a pedagogical tool, were intricately designed to address the research inquiries while concurrently developing a computational thinking assessment instrument. A systematic analysis addressed the research problem formulation after the data collection phase. Examination of test outcomes yielded insights relevant to the first research question, while detailed classroom observations and analyses were undertaken to elucidate the impact of Scratch programming on students' understanding of geometric principles, explicitly addressing the second research question.

Schools' and Students' Target

The research was conducted at one of the public primary schools in Ibaraki prefecture, Japan, involving a total of 99 sixth-grade students. It is known for its commitment to providing quality education to its students. The school is situated in a suburban area, providing a conducive learning environment with modern facilities and resources. The student body at this school is diverse, comprising students from various socio-economic backgrounds and with differing levels of academic abilities.

In terms of the students' characteristics, they are typically between the ages of 11 and 12 years old, representing a crucial developmental stage in their academic journey. Sixth-graders transition from primary to secondary education, often requiring increased academic rigor and responsibilities. The students are known for their diligence, enthusiasm for learning, and respectful demeanor toward teachers and peers. On the other hand, the research in Indonesia was conducted at one of the private schools in Yogyakarta, involving 28 fifth-grade students. This school is a reputable educational institution affiliated with the Muhammadiyah organization, which is well-known for its contributions to education in Indonesia. Located in a bustling urban area, the school serves a diverse student population from various cultural and socio-economic backgrounds.

The characteristics of its students are similar to those in Japan, with fifth-grade students typically aged 10 to 11 years old. They are at a crucial stage in their academic development, preparing for the transition to higher levels of education. These students are known for their resilience and eagerness to learn despite facing challenges such as limited resources and socio-economic disparities. Overall, both schools provide unique learning environments that contribute to their students' academic and socio-emotional growth. The diverse student populations and dedicated faculty members create dynamic educational settings conducive to research and innovation in teaching and learning practices.

Intervention Phase of Lessons

The study was designed to cultivate foundational computational thinking abilities over the course of three instructional sessions, with the objectives and content outlined in Table 1. The initial lesson concentrated on acquainting students with the fundamental principles of algorithmic thinking, covering sequential processing, iteration, and conditional branching, drawing from works by Shute et al. (2017) and Rose (2019). This introduction was facilitated through an innovative integration of dance and physical engagement. Subsequently, Scratch was introduced as a platform to apply these concepts tangibly, exemplified by generating squares through sequential and iterative processes. The second lesson focused on creating various regular polygons, with students actively determining the precise angles at each vertex and the requisite number of iterations for each shape, thus enhancing their comprehension. The third lesson aimed to consolidate the knowledge acquired from preceding sessions, challenging students with a more intricate task—drawing star shapes—requiring the application of their understanding in a nuanced context. Additionally, students underwent assessments to gauge the development of their computational thinking skills, with evaluation aimed at measuring the internalization of computational thinking principles demonstrated through adept application across diverse and progressively intricate scenarios.

Table 1. Objectives and Content of the Lessons

Lesson	Objectives	Content
1	- Understand the basic usage of Scratch. - Draw a square using sequential and iterative processing.	- Experience sequential processing, iterative processing, and conditional branching, which are key concepts in algorithmic thinking, through physical activities. - Draw a square using sequential and iterative processing.
2	Draw various regular polygons.	- Think about a method to draw a triangle using iterative processing. - Draw a square, pentagon, and circle.
3	- Draw a star. - Using the concepts learned so far, draw the shapes provided (as an assessment).	- Utilize the knowledge acquired so far to devise a method for drawing a star. - Draw the shapes presented (Levels 1-8). - Draw shapes using free imagination.

Assessment and Analysis Method

To formulate the assessment task, several prior studies were consulted for guidance on assessment methods (Basso et al., 2018; Román-González et al., 2019), educational interventions, and strategies for teaching and evaluating computational thinking (Rose, 2019; Bender et al., 2023), particularly in the context of programming with Scratch. The assessment task comprised eight figures ranging from Level 1 to Level 8, which students were required to draw using Scratch programming. The figures corresponding to each level are illustrated in Figure 1, and a time limit of 20 minutes was allocated for this activity.

Computational Thinking (CT) comprises four key components: decomposition, pattern recognition, abstraction, and algorithmic design (Wing, 2006; Kalelioğlu et al., 2016; Yunianto et al., 2023; Purwasih et al., 2024). Decomposition entails breaking down a complex problem into smaller, more manageable parts or subproblems. Pattern recognition involves identifying patterns, similarities, or regularities within the problem or data. Abstraction involves focusing on essential details while filtering out unnecessary information. Algorithmic design entails developing step-by-step instructions or procedures to solve problems. Criteria for evaluating each task were established based on these definitions. Despite tasks being developed up to Level 8, students only responded up to Level 3.

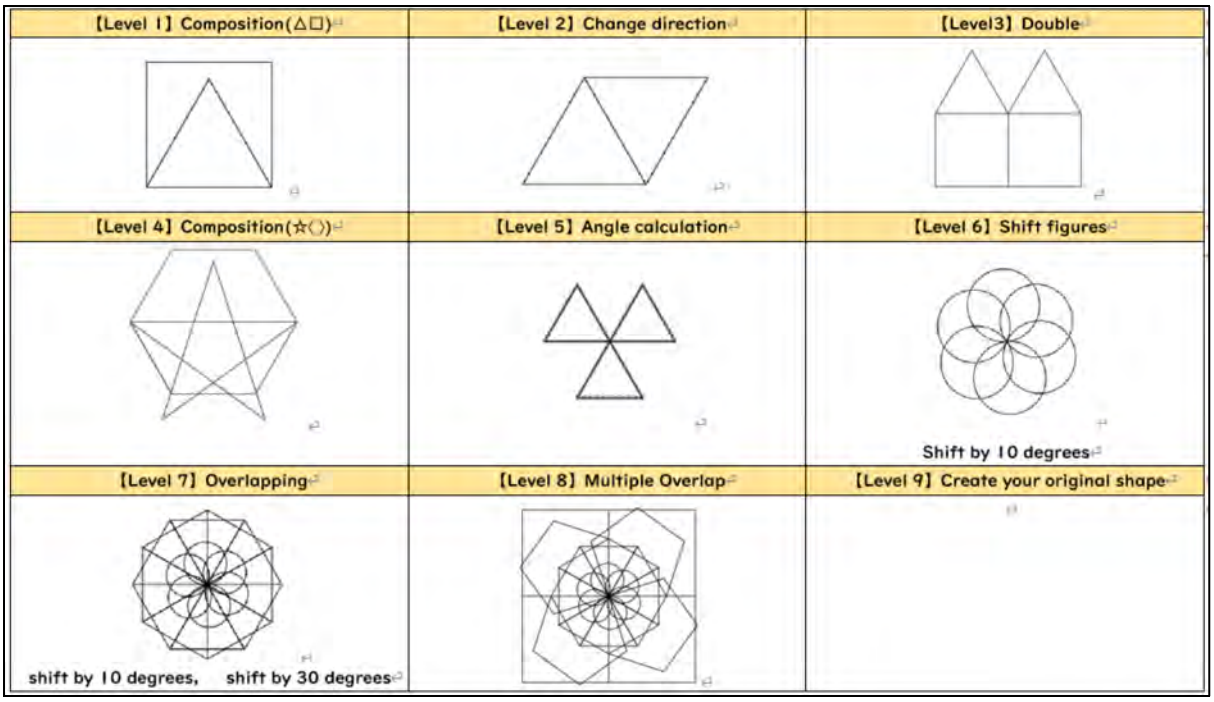

Figure 1. Task Assessment

Therefore, an evaluation table is provided for up to Level 3 (Table 2). Moreover, the assessment was carried out for Decomposition, Pattern Recognition, and Abstraction, concentrating on whether these facets were accomplished as per the established criteria. For example, concerning Level 1, if criterion 'a' was fulfilled, it was inferred that Decomposition thinking was demonstrated. As for Algorithm Design, outcomes were classified based on whether sequencing, looping, or both were employed.

Table 2. Evaluation Criteria for Computational Thinking (CT)

Computational Thinking (CT)	Achieved or Not	Level 1	Level 2	Level 3
Decomposition	✓	a. Draw a square and a triangle. b. Other ways	a. Draw two triangles b. Other ways	a. Draw all four figures separately b. Draw two triangles separately and squares in other ways c. Draw two squares separately and triangles in other ways d. Other ways
	☐	The triangle fit inside the square without overlapping	a. Using two triangles b. Drawing sides by sides with some	a. Drawing square → square → triangle → triangle b. Drawing triangle →

			patterns	triangle → square → square c. Drawing triangle → square → triangle → square d. Drawing square → triangle → square → triangle
Pattern Recognition	✓	a. Drawing a square and a triangle consecutively b. The triangle does not fit inside the square without overlapping	Draw two triangles consecutively	Drawing squares and triangles consecutively
	□	Drawing by other ways	Other ways	Drawing by other ways
Abstraction	✓	Drawing a square and a triangle consecutively	Draw two triangles consecutively	Drawing squares and triangles consecutively
	□	Drawing by other ways	Other ways	Drawing by other ways
Algorithm Design	✓	a. Use "Sequence" to draw figures b. Use "Loops" to draw a triangle c. Mixed	a. Use "Sequence" to draw figures b. Use "Loops" to draw a triangle c. Mixed	a. Use "Sequence" to draw figures b. Use "Loops" to draw a triangle c. Mixed

RESULTS AND DISCUSSION

Scratch for Beginners

Overall, the students demonstrated a general understanding of the three core concepts of algorithmic thinking. Integrating physical activities, visual comprehension using Scratch blocks, and verbal explanations appeared to be effective strategies. However, it became evident that the terminology posed challenges for the students. Consequently, efforts were made to simplify the language as much as possible from the second lesson onwards, employing plain Japanese terms such as "processing in order," "repetition," and "condition." This adjustment aimed to enhance clarity and facilitate better comprehension among the students.

Using Scratch, the sample display on the monitor and the worksheet provided visible references for the students to create programs mimicking and drawing squares. However, due to the abundance of blocks available, it proved challenging for students to locate the necessary ones. To address this, a project was prepared in which the required blocks were pre-arranged separately. The activity was structured step-by-step, guiding students through the sequential processing and repetition required to draw squares. This approach facilitated smooth operation and enhanced learning outcomes. While some students had prior experience with Scratch, it was their first exposure to the platform for many. Consequently, the activities generated significant interest among the students, regardless of their prior familiarity with Scratch.

In the initial session of the Indonesian class, students enthusiastically participated in the structured lesson plan, which spanned 45 minutes and comprised three distinct activities: introduction (10 minutes), expansion (30 minutes), and consolidation (5 minutes). To augment algorithmic thinking, the physical activities introduced in the Japanese class were incorporated into the Indonesian class. Three specific physical exercises were integrated to aid students in comprehending Scratch, emphasizing the importance of adhering to a process, persisting until conditions are met, and adjusting strategies based on varying conditions.

During the expansion phase, teachers elucidated the lesson's objective to students: "to manipulate blocks to program something with Scratch." Teachers demonstrated the process for students, illustrating how to draw a square using 'go' and 'turn' blocks, as shown in Figure 2. This step was pivotal in the initial lesson, fostering computational thinking by delving into the functionalities of sequencing, loops, and selections within the Scratch activity.

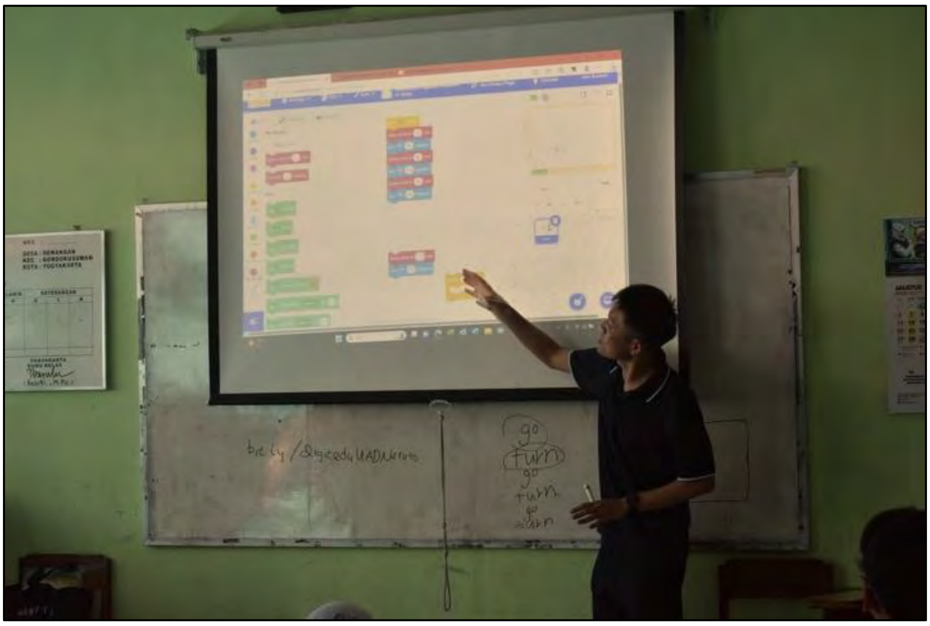

Figure 2. The Teacher Simulated How to Draw a Square using 'Go' and 'Turn' Blocks

In the activities of this initial lesson, students not only attempted to draw using block actions in the Scratch program but also grasped the intricate relationship between loops utilizing the repeated function and the number of sides of the shape to be created. This understanding benefitted students as they tackled other spatial figures in subsequent lessons. In the final minutes of consolidation, students were encouraged to reflect and summarize their comprehension of sequence, loops, selections, and how to manipulate blocks within Scratch. This reflective exercise facilitated a deeper internalization of the lesson's key concepts and encouraged students to articulate their newfound understanding.

Computational Thinking Skills in Solving Geometry Problems

During the triangle drawing activity, students demonstrated the application of knowledge acquired in their math class. We physically moved the sprite illustration on the blackboard to confirm the procedure for drawing triangles as a whole. Notably, when considering the degree of rotation required after drawing a straight line, many students initially thought "60 degrees" instead of the correct answer, "120 degrees." Upon further inquiry, students cited concepts such as "sum of the interior angles of a triangle" and "one angle of a triangle" from their math lessons. This exemplified their utilization of mathematical knowledge within the programming context. While the sprite's actual rotation of 60 degrees was displayed on the monitor, students quickly recognized the discrepancy. They found it relatively straightforward to determine the number of repetitions for the straight line and 120-degree rotation, likely drawing on their prior experience from creating squares in the previous lesson. Consequently, by amalgamating mathematical knowledge with programming concepts learned in the initial lesson, students successfully understood and replicated the triangle drawing program using Scratch.

In the activity of drawing pentagons and circles, students needed help in calculating the required angles. However, they independently tackled the problem through trial and error, seeking peer assistance when needed. After some time, we provided the angles which enabled most students to complete the program successfully. Those who had already formed pentagons and circles drew polygons with six or more sides, referring to the suggested angles, as presented in Figure 3.

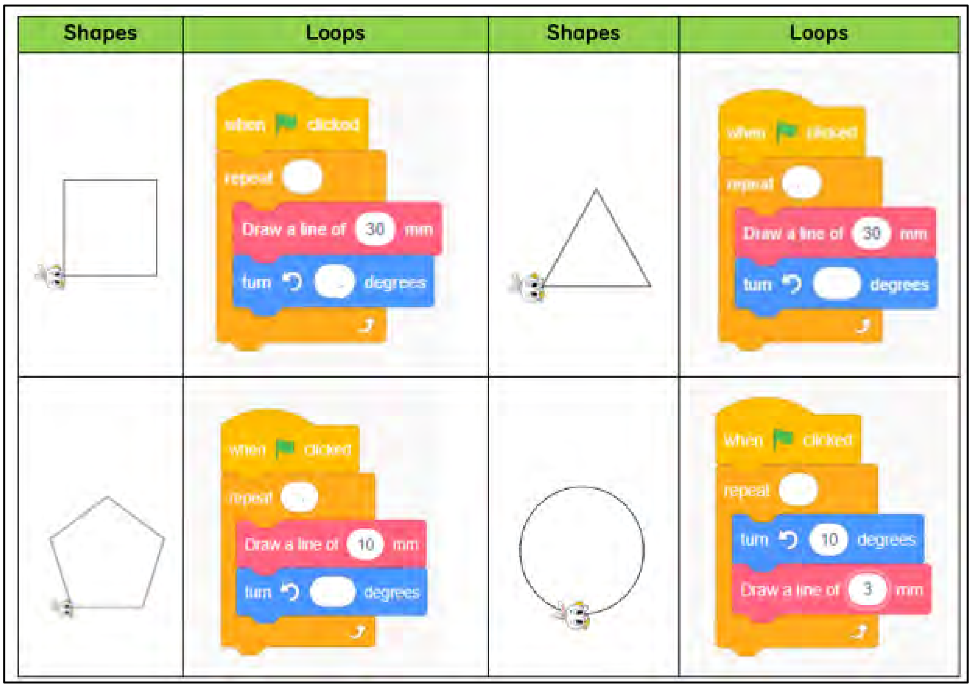

Figure 3. Students' Worksheet Instruction

During the reflection presentation, one student remarked, "I noticed that as the number of corners increased, the figure gradually got closer to the circle." This trend observed in the Indonesian classroom aligns with the experiences in Japan. In the second lesson in Indonesia, teachers review the previous lesson and reiterate to students the three essential functions of sequence, loops, and selections. During the expansion phase, the lesson's objective is introduced: by the end of the class, students should be capable of drawing various figures such as triangles, squares, pentagons, and circles using Scratch.

To facilitate the investigation, students are presented with figures in their worksheets and tasked with determining the number of sides in each figure, the degrees in each corner, and the repeated actions involved. While Indonesian students find the first shape, the 'rectangle,' relatively straightforward due to prior practice, they encounter challenges with triangles and subsequent shapes. Teachers guide students in comprehending the total degrees in each shape, addressing considerations such as turn degrees that may lead to confusion. Students employ a trial-and-error strategy, experimenting with various angles within the Scratch program. Some students, like Shaina, grasp the concept of supplementary angles and successfully construct shapes like triangles, rectangles, pentagons, hexagons, polygons, and circles, as illustrated in Figure 4.

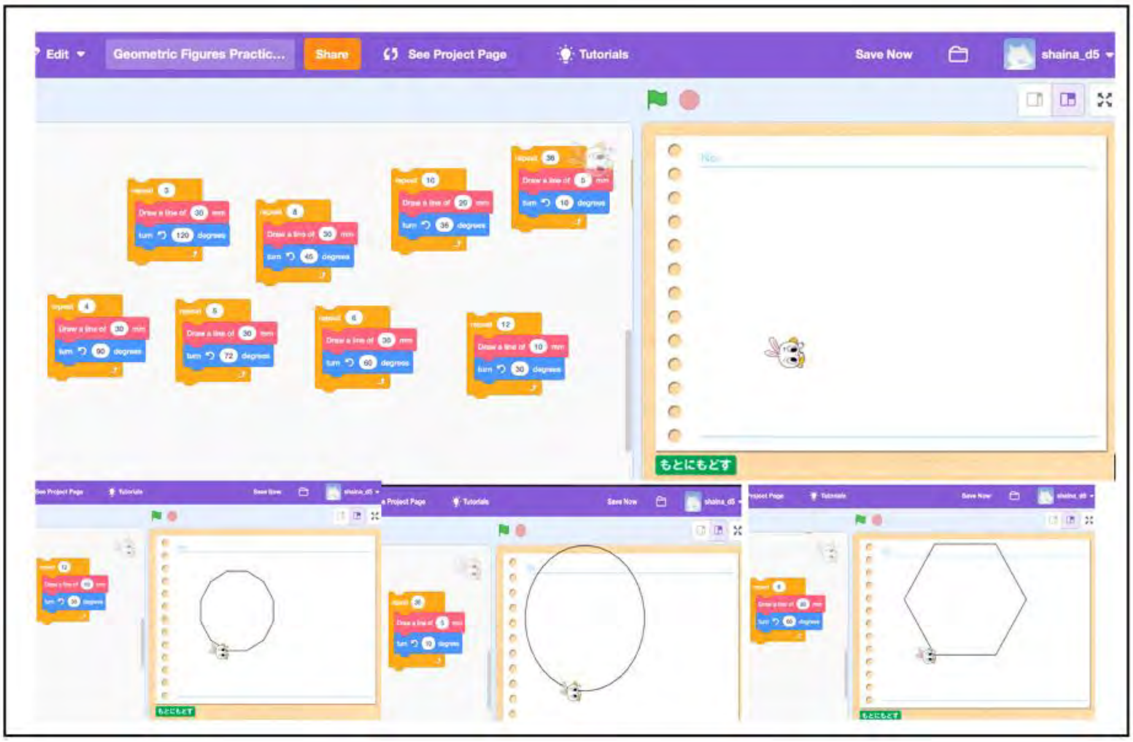

Figure 4. Shaina's Work in Different Shapes

Understanding the values of exterior angles of regular polygons is imperative for students' mathematical comprehension. To achieve this, students employed three primary methods. Firstly, they used a protractor to measure the exterior angles of regular polygons inscribed on paper, presented in Figure 5.

Figure 5. Finding the Value of Exterior Angles Using a Protractor

Secondly, students calculated the exterior angles of regular polygons in their notebooks before coding, as referred to in Figure 6. Lastly, students engaged in an iterative process of experimentation, employing trial and error to refine their understanding of exterior angles through practical application.

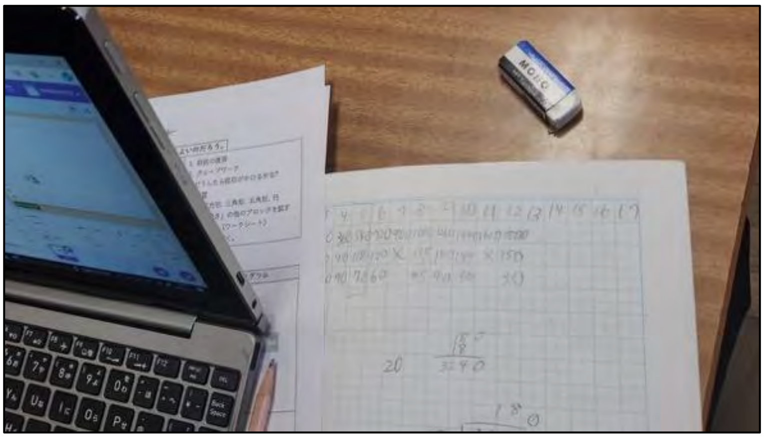

Figure 6. Finding the Value of Exterior Angles by Calculation Beforehand

Subsequent figures showcase students' approaches to creating circles, highlighting instances of manipulating repeated actions or adding additional Scratch program blocks (forever blocks) to achieve the desired result, as shown in Figure 7.

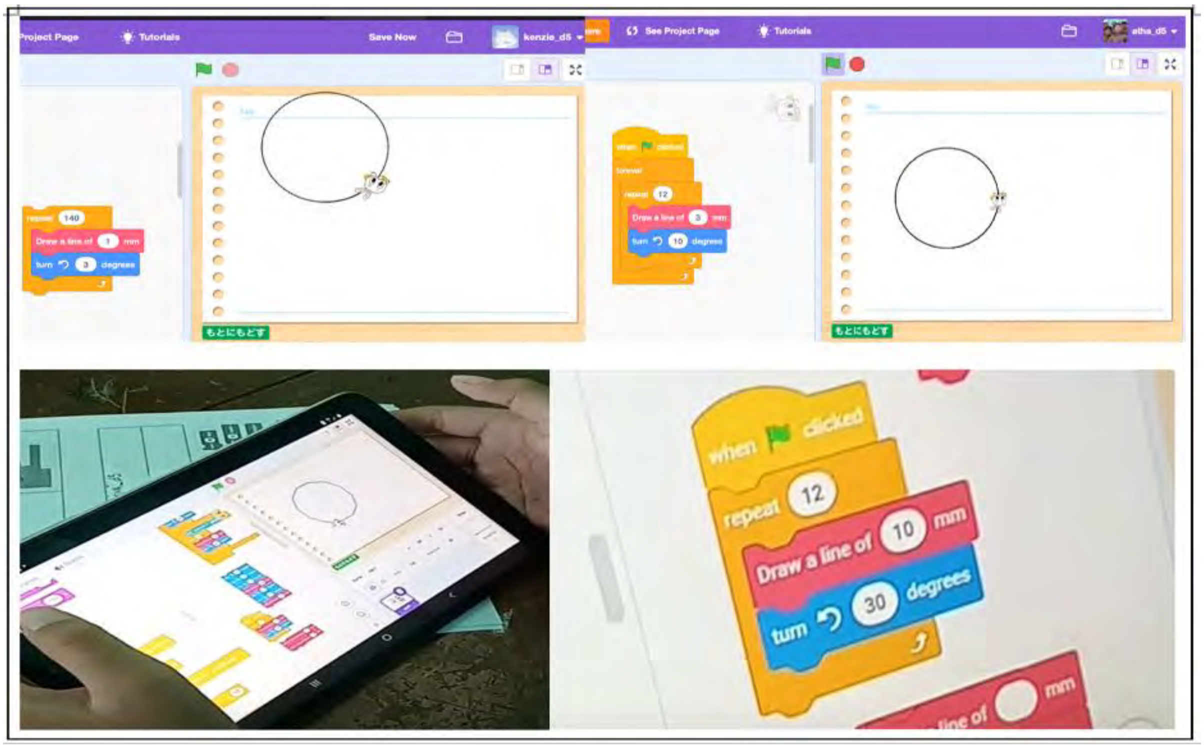

Figure 7. Some Students Work using Different Ways

This activity serves as a catalyst for fostering students' cognitive processes in shape construction, placing particular emphasis on three pivotal factors essential to the Scratch program: side, angle, and action. By comprehending the degrees present in each corner and accurately inputting the corresponding angles, students gain the ability to create their own works in subsequent lessons. Furthermore, while computing exterior angles of regular polygons may not immediately verify their correctness, utilizing

Scratch to draw shapes offers instantaneous confirmation. The fact that nearly all students successfully drew up to a regular hexagon indicates that using Scratch effectively enhances their understanding of geometry.

Exploring Students' Computational Thinking Skills through Geometry Problems

In the final lesson, we reviewed the sequential processing and repetition program with a task involving drawing a star. As the activity revolved around presenting angles, students could develop a program swiftly using familiar techniques.

Each student uniquely approached the problem in the level-specific tasks, employing one of three primary methods: trial and error, pre-calculation, and mathematical tools. Trial and error involve predicting approximate angles and movements and then testing the program iteratively. While this method is time-consuming and requires patience, inadvertently applying shapes from other levels is risky. Pre-calculation, conversely, entails pre-planning calculations and angle movements on a worksheet, facilitating the organization of thoughts before programming. This approach minimizes the likelihood of significant errors due to the necessary preparation involved. Finally, the mathematical tools method involves utilizing tools such as a protractor or ruler to analyze shapes on the worksheet before programming. This method, prevalent in Japanese education, ensures students can access the necessary mathematical exploration tools.

During the 20-minute activity session, the completion rates of tasks varied among Japanese students across different difficulty levels: 48 out of 99 students completed Level 1 tasks, 42 completed Level 2 tasks, and 16 completed Level 3 tasks, with each number denoting the portion of students out of the total of 99 who completed the task. The lesson conductor and the homeroom teacher providing support observed sustained engagement among most students throughout the activity. Notably, many students, including those typically less inclined towards mathematics and those enrolled in special needs classes, demonstrated focused concentration on their tasks. This heightened engagement was attributed to the efficacy of the ICT equipment and the Scratch platform itself. Furthermore, during the final 10 minutes of the session, students were encouraged to express their creativity by drawing their own shapes, further enhancing their involvement in the activity.

Figure 8. Teachers Explain How to Draw a Star

On the other hand, the concluding session in the Indonesian class serves as the apex of the course, centering on the acquisition of the skill of drawing a star under the guidance of instructors, as shown in Figure 8. Subsequently, students are allocated time to review and refine their learned figures. Following this, they engage in challenging tasks involving the creation of various shapes, with eight levels of complexity delineated as challenges, as illustrated in Figure 1. Each level necessitates the application of distinct computational thinking for successful completion. The teacher evaluates students' computational thinking skills and achievements, as presented in Table 2. Furthermore, students undertake the endeavor of devising original shapes, showcasing their unique ideas and algorithmic thought processes in Level 9 of the assigned task.

During the introductory phase, teachers elucidate to students the objective of the day's lesson: to acquire the technique of drawing a star and evaluate their algorithmic thinking skills by the lesson's conclusion. Additionally, students are reminded of the concepts covered in previous lessons to provide context for the current material. Before progressing to the expansion phase, students participate in a review and practice session under the teacher's guidance. This session focuses on drawing various figures and angles, employing blocks, and adhering to programming naming conventions.

Teachers guide students in drawing a star using both sequential and loop-based methods, illustrating the efficiency of loops compared to the traditional sequential approach. Students are prompted to contemplate the number of lines required to form the star and the corresponding number of blocks necessary for its creation, as shown in Figure 9. Emphasizing the star's angle, set at 144 degrees, the teacher aims to deepen students' understanding of geometric concepts. The primary goals of this activity encompass two aspects: firstly, to draw the star ("*Bintang*" in the Indonesian language) using both sequential and loop methods, and secondly, to incorporate turn left and turn right blocks, laying the groundwork for upcoming challenges.

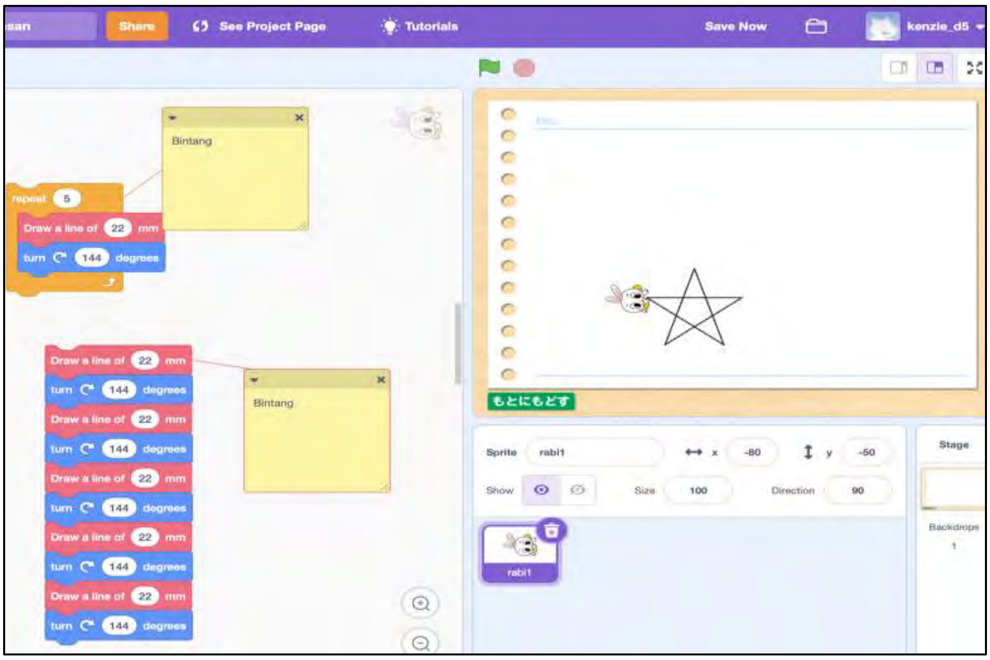

Figure 9. Student's Work in Making Stars

The provided figure exemplifies the outcomes of students' endeavors in generating star patterns through sequential and loop techniques, employing turn-right blocks. Notably, students discern that fashioning a star necessitates five lines, correlating with the 144-degree angles previously elucidated by the teacher. In constructing the elongated rendition of the star via the sequential method, students adeptly replicate the process by duplicating the red (signifying drawing a line) and blue (denoting a turn right) blocks five times. Furthermore, drawing from their prior experiences, students recognize the efficiency of employing loops and inputting '5' in the repetitive function, significantly expediting the creation of star shapes.

Five minutes before transitioning into the expansion phase, teachers delineate the guidelines for tackling the challenging task outlined in their worksheets, which include:

- A time constraint of 20 minutes was allocated for levels 1 through 8.
- Mandatory naming of programming blocks upon completion.
- Prohibition against breaking completed blocks; students should duplicate blocks if necessary.
- Prohibition against seeking answers from peers.

Within the allotted 20-minute timeframe covering levels 1 through 8 of the challenging tasks, as shown in Figure 1, most students successfully advanced up to level 3. Two predominant approaches emerged among the various strategies employed in their construction processes: trial and error and design simulation. Students employing the trial-and-error strategy explore the program by iteratively adjusting angle inputs, turning left/right blocks, and other parameters. Conversely, students utilizing the design simulation approach first sketch the requested shape in their notebooks, calculate the angles, and then replicate the process in the program. Many students emphasize a combination of sequences and loops when designing the requested shapes.

Level 1 tasks students with creating a triangle and a square, as shown in Figure 10. They are assigned to draw the specified figures and label the programming blocks as 'Level 1,' utilizing sequences, loops, or devising their own method for constructing the shape.

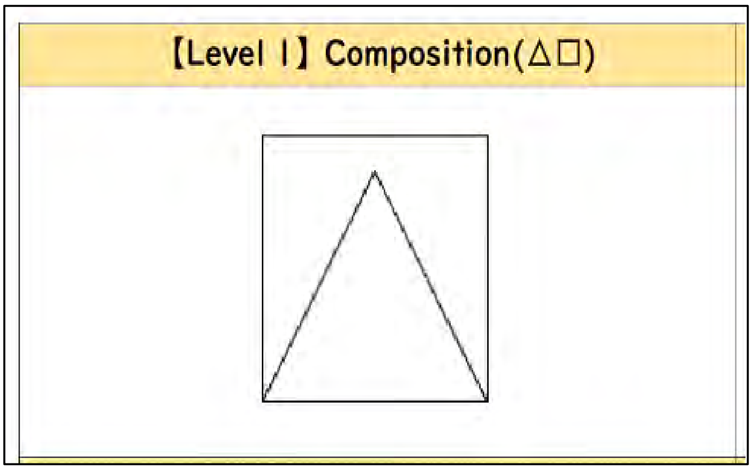

Figure 10. Level 1 - Composition of Triangle and Square

Out of the 28 Indonesian students, 19 demonstrated proficiency in drawing using various approaches. Among them, fifteen students met the evaluation criteria for computational thinking presented in Table 2, encompassing decomposition (DC), pattern recognition (PT), abstraction (AB), and algorithmic design (AD). Conversely, four students solely met the algorithmic design (AD) criteria. Sample examples showcasing students reaching all phases within computational thinking skills involve creating shapes of squares and triangles that either do not overlap (see Figure 11) or overlap with each other (see Figure 12).

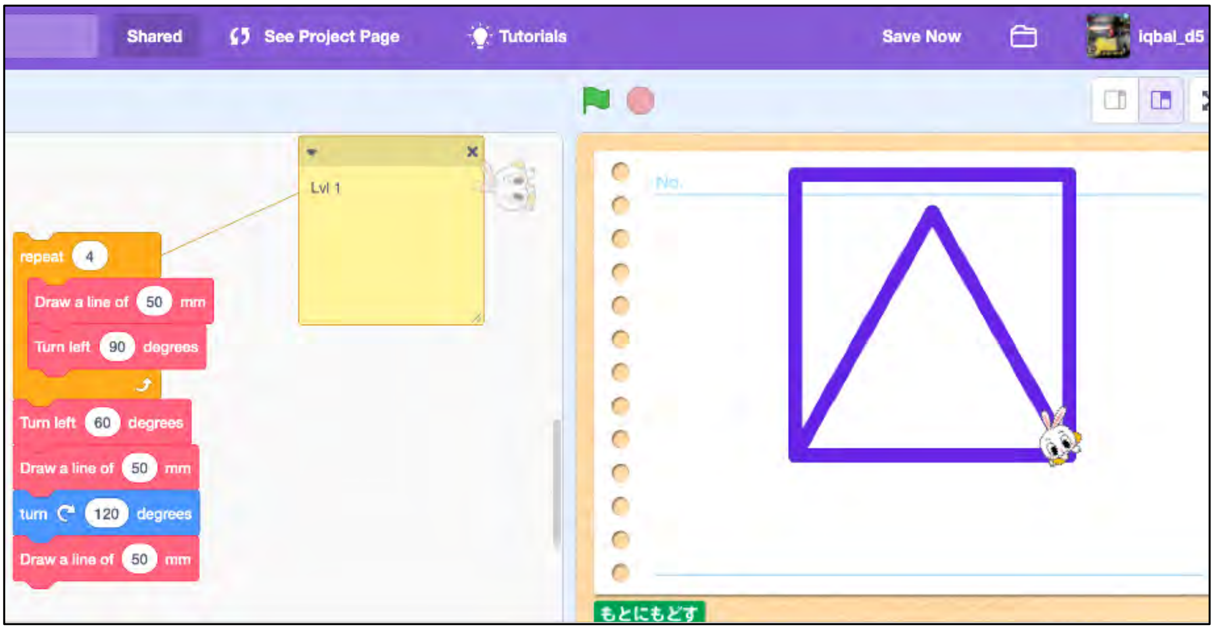

Figure 11. Student's Work in Level 1 - With Category Shapes Not Overlapping in DC

In contrast, in Japan, concerning Level 1, 48 out of 99 students have fulfilled the four criteria of computational thinking. Interestingly, regarding algorithmic design, only five individuals constructed using sequences, while the remaining 41 utilized loops and two students used mixed methods (both sequence and loop). Figures 11 and 12 highlight discrepancies in strategy usage: the student depicted in Figure 11 utilizes loops to outline the rectangle and subsequently employs sequences to delineate the triangle, ensuring there is no overlap between the two shapes. Conversely, the student in Figure 12 utilizes loops for both shapes. They input '6' into the repeated blocks to construct the rectangle, facilitating the accurate initiation of the triangle drawing process.

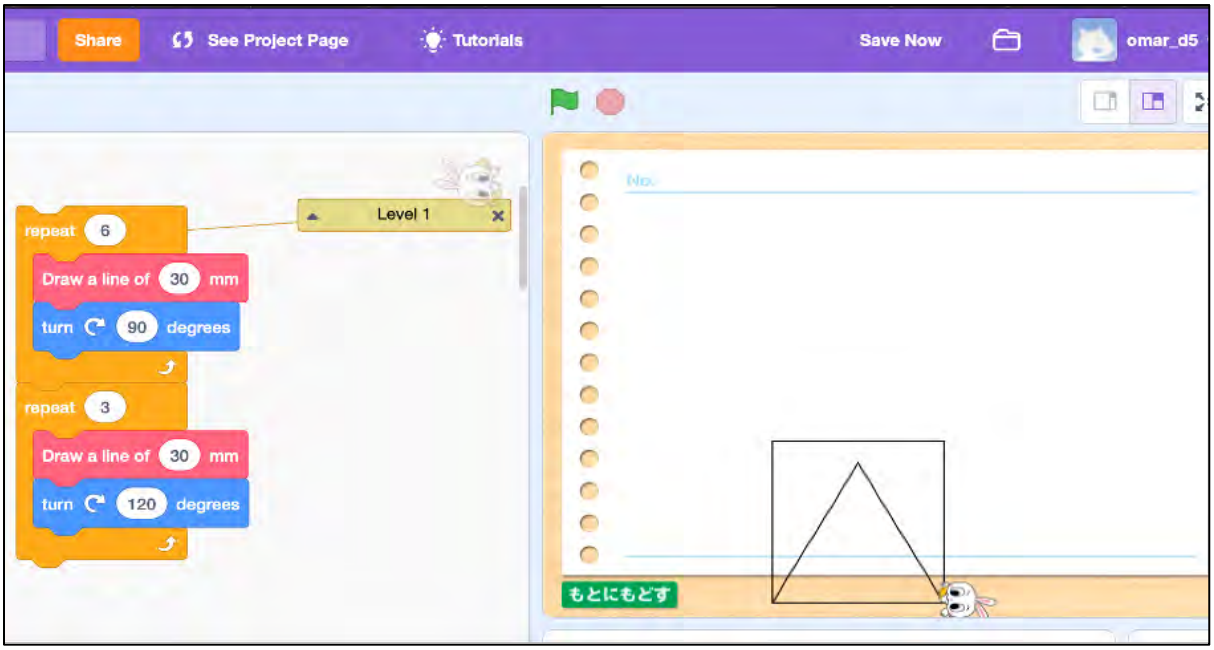

Figure 12. Student's Work in Level 1 - With Category Shapes Overlapping in DC

Both students successfully fulfill the decomposition and pattern recognition criteria while drawing two distinct shapes. Despite differences in abstraction, the student depicted in Figure 11 adeptly avoids overlap and can draw the two shapes consecutively, meeting criterion a. This student also demonstrates algorithmic design (criterion c) by employing a combination of loops and sequences.

Conversely, the abstraction ability of the student in Figure 12, who creates a rectangle by overlapping before directly drawing a triangle, aligns with criterion b. Additionally, this student meets criterion b in algorithmic design, utilizing only loops to craft the requested shape in level 1.

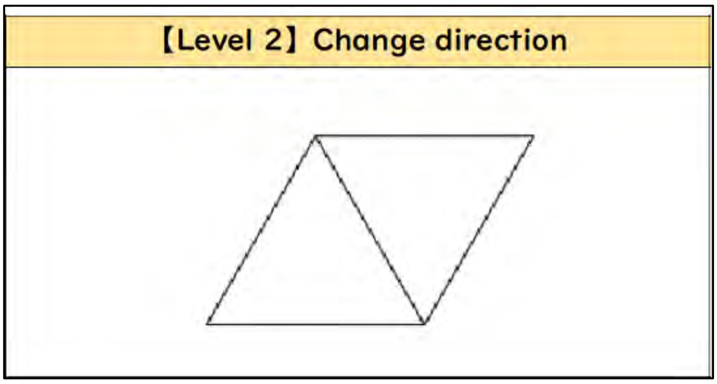

Figure 13. Level 2- Change the Direction of Two Triangles

In Level 2, students are tasked with altering the orientation of two triangles, as shown in Figure 13, requiring them to draw them in opposite orientations using sequences, loops, or a combination of strategies. Among the 28 Indonesian students, eight successfully met the level criteria. Among them, four students achieved all the criteria of Computational Thinking (CT), while another four met only one of the CT criteria, specifically algorithmic design (AD).

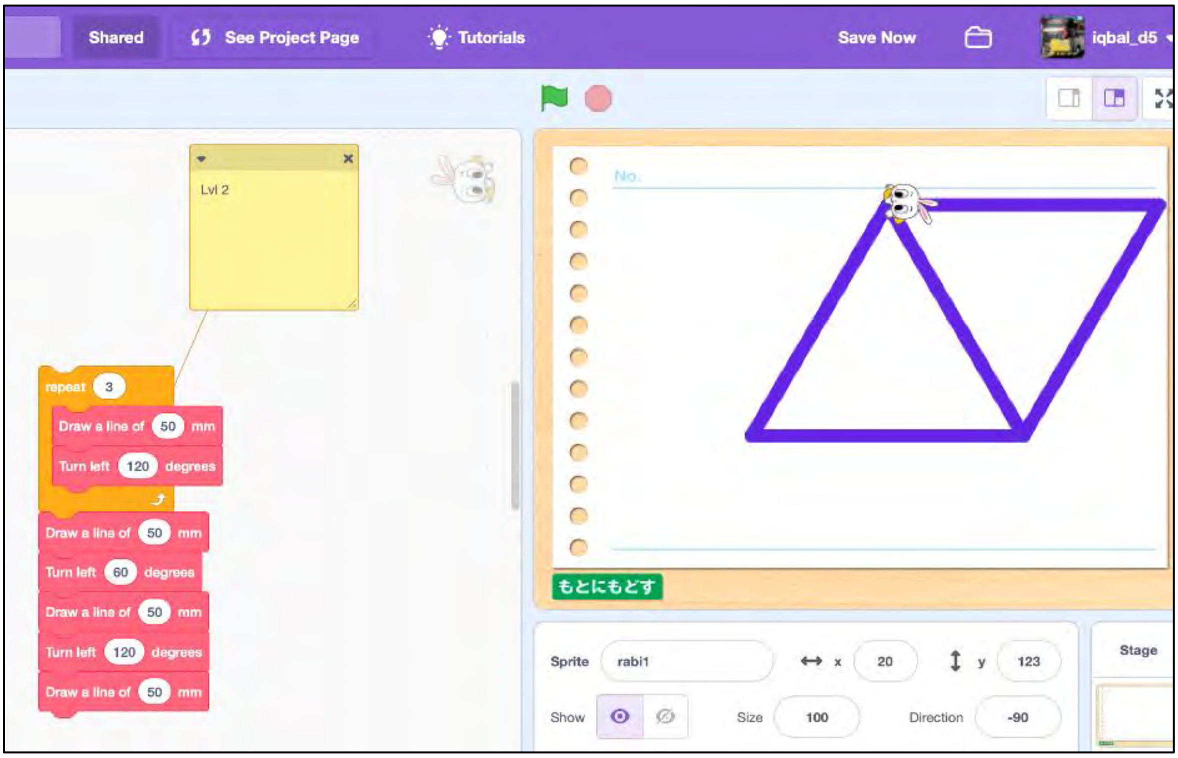

Figure 14. Student's Work in Level 2 - With Category Side by Side, Overlapping, and Mix

Conversely, 42 out of 99 students in Japan achieved all the computational thinking goals. Sample examples illustrating students reaching all phases of computational thinking skills (DC, PT, AB, AD) include creating shapes of two triangles, whether overlapping or not, drawing side by side or using two triangles, and employing mixed algorithmic design.

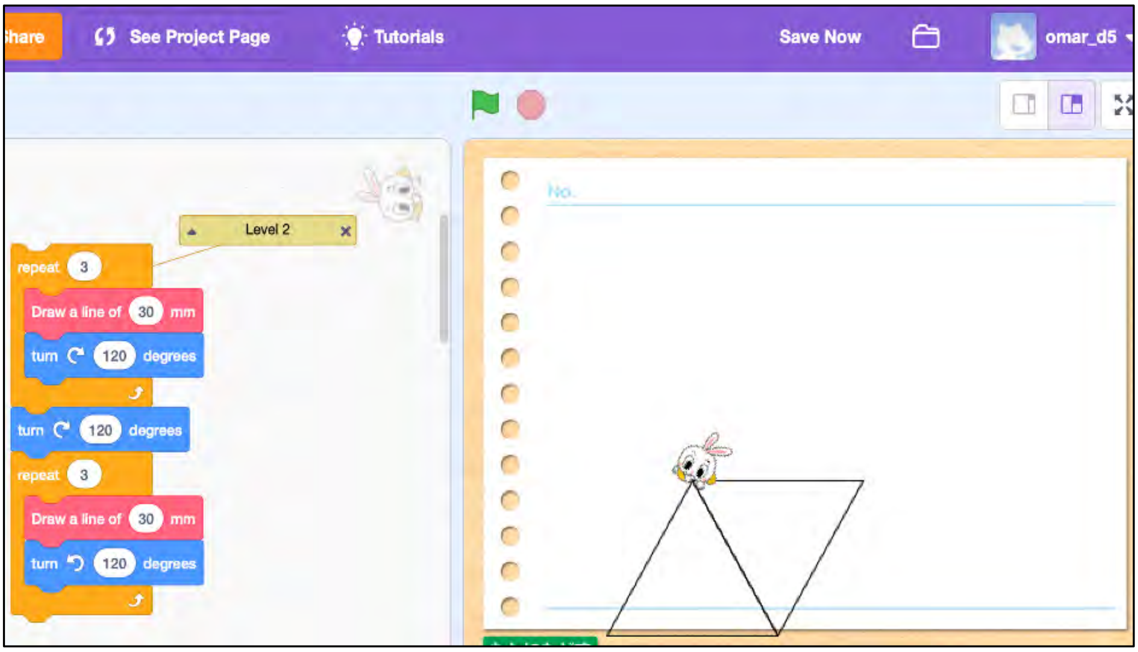

Figure 15. Student's Work in Level 2 - With Category Two Triangles, Overlapping, and Loop

Figures 14 and 15 exemplify the fulfillment of decomposition (DC) criteria and the variations in composing two triangles in different orientations. Figure 14 showcases the arrangement of two triangles side by side, demonstrating pattern recognition (PT) criteria "b," while Figure 15 depicts the utilization of two triangles fulfilling PT criteria "a." In Figure 14, the student employs a mixed algorithmic design (AD), incorporating loops and sequences (criteria c) to construct the two triangles. Conversely, in Figure 15, the student initially draws a triangle upside down and then completes the shape by drawing the second triangle using loops only (criteria b). Both students adopt distinct approaches to abstraction (AB): the student in Figure 14 requires an additional line to continue drawing the next triangle shape (criteria b), while the student in Figure 15 draws two triangles consecutively (criteria a).

Level 3 tasks students with duplicating the given shape using sequences, loops, or their unique methods (refer to Figure 16).

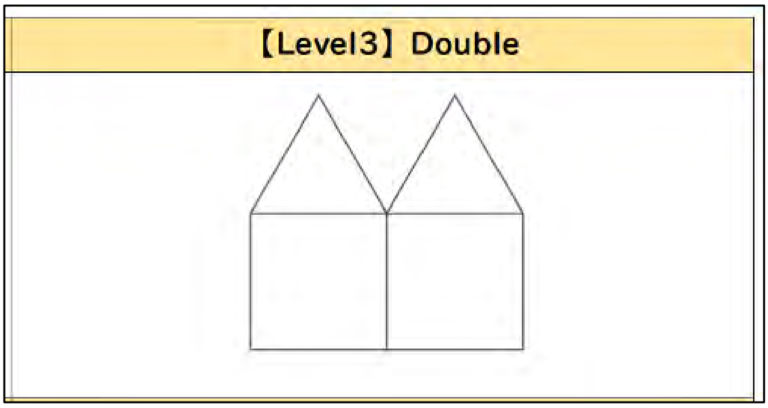

Figure 16. Level 3 - Double Shape

Out of the 28 Indonesian students, only four successfully met all criteria in Computational Thinking (CT). Three students fulfilled criteria 'a' in decomposition (DC) by drawing all four figures (two triangles and two squares) separately, while one student used an alternative method, meeting criteria 'd' of DC. In pattern recognition (PT), one student fulfilled criteria 'a' by drawing square → square → triangle → triangle, another met criteria 'c' with triangle → square → triangle → square, and two students met criteria 'd' by drawing square → triangle → square → triangle. Moreover, all four students consecutively adhered to the 'a' criterion in abstraction (AB) by drawing squares and triangles. In algorithm design (AD), one student met criteria 'b' by using 'loops' to draw a triangle, and three students met criteria 'c' by incorporating a mix of sequences and loops.

In Japan, 16 out of 99 students achieved all the goals of computational thinking. Concerning algorithmic design, 15 students used sequences, while only one utilized a loop. Regarding PT, only six students divided the shapes into triangles and squares, whereas the other ten did not confine themselves to geometric shapes and notably mixed in straight lines in their drawings. The accompanying figures showcase samples of students' work that encompass all phases of computational thinking skills (DC, PT, AB, AD), as shown in Figure 17.

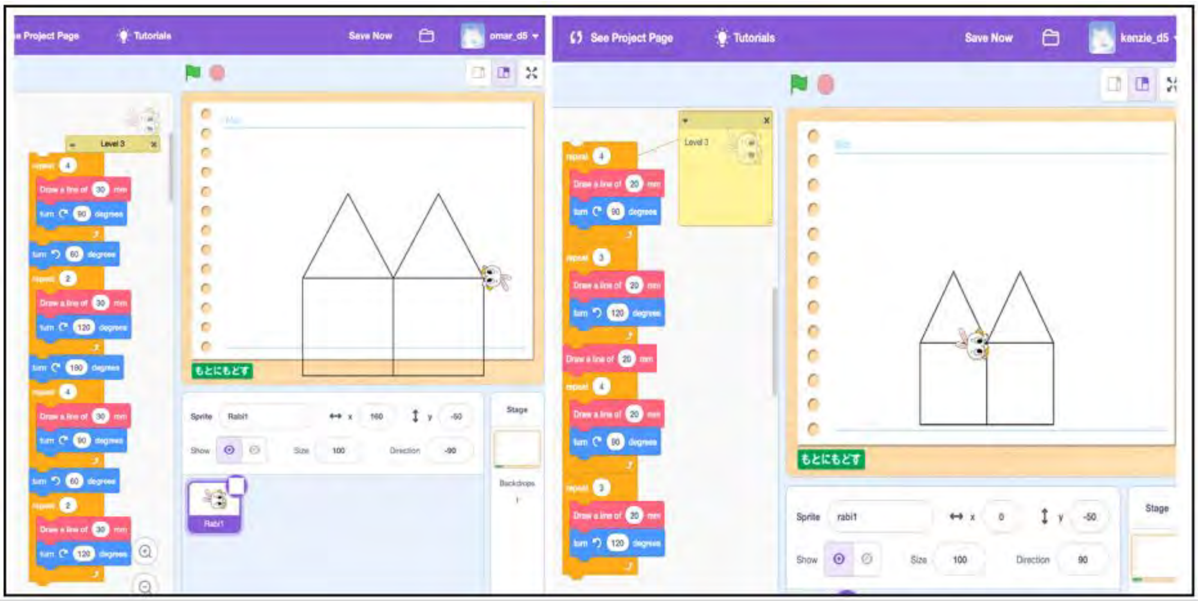

Figure 17. Students Work in Level 3

In summary, both students exemplify similar criteria in various aspects of computational thinking: 'a' in decomposition (DC), 'd' in pattern recognition (PT), 'a' in abstraction (AB), and 'c' in algorithm design (AD). However, notable differences emerge upon inspecting the students' programs, as shown in Figure 17. The program of the left student appears relatively lengthy (11 blocks), and the input for the triangle shape in the repeated block is not 3 but 2. The left student diligently aims to minimize overlap among all four shapes by adjusting the direction by turning degrees before drawing the next shape. Conversely, the program of the right student is shorter (9 blocks), yet it results in multiple lines overlapping, as presented in Figure 17.

Overall, we did the analysis presented in Table 3, which provides a comprehensive comparison of computational thinking skills among students in Japan and Indonesia, highlighting their abilities at different proficiency levels. The study focuses on four key components of computational thinking: Decomposition (DC), Pattern Recognition (PT), Abstraction (AB), and Algorithmic Design (AD). These components are crucial in understanding how students approach and solve complex problems using computational methods.

Table 3. Computational Thinking Skills Students in Indonesian and Japan

No	Group	Students Characteristics	Computational Thinking Skills (Scratch Programming)		
			Level 1	Level 2	Level 3
1	Japan	Grade 6 Primary	98.44 %	75 %	79.69 %
2	Indonesia	Grade 5 Primary	69.74 %	59 %	93.75 %

The data is meticulously segmented into three levels, each representing varying degrees of proficiency in essential skills, allowing for a nuanced understanding of students' capabilities at different stages of their learning journey (Ogegbo & Ramnarain, 2022). Furthermore, this segmented data outlines the percentage of students achieving each level in both countries, offering detailed and comparative insights into the computational thinking abilities of students in these regions. In Japan, over 70% of students have achieved each level, whereas in Indonesia, while Level 2 remains at around 60%, Level 3 indicates a value of over 95%. These results suggest that the classes designed in this study effectively fostered computational thinking. Such a comparison is crucial for assessing the effectiveness of the educational approaches adopted in Japan and Indonesia, providing a clear picture of how students in these countries are being prepared to navigate a world increasingly dominated by technology and digital solutions (Ausiku & Matthee, 2023).

To address the research question regarding the extent to which the designed learning promotes the computational thinking skills of elementary school students in Japan and Indonesia, the results of lesson observations from day one to three indicate significant progress. Students achieved all the indicators of computational thinking as outlined in the lesson plans developed for the study. Their ability to draw geometric figures using Scratch evidenced their enhanced computational thinking and understanding of polygons. Throughout the course, students comprehended the concept of algorithmic thinking, mastered basic operations in Scratch, and successfully drew polygons. The effectiveness of understanding algorithmic thinking can be attributed to three key factors: physical activity, visual comprehension, and experiential learning using Scratch. Particularly noteworthy is the role of hands-on, interactive learning experiences in deepening students' comprehension and engagement with computational concepts, consistent with prior research findings (Aminah et al., 2023; Piedade & Dorotea, 2022; Rafiepour & Farsani, 2021; Jiang & Li, 2021; Yunianto et al., 2023).

Furthermore, the second research question regarding the extent to which the designed learning enhances students' understanding of geometry based on their computational thinking skills is evident in the student's work throughout the program, particularly in lesson three, where they completed the assessment in Figure 1. A notable aspect of our intervention was using Scratch programming to facilitate geometry learning. Students in both countries exhibited improved abilities to create geometric shapes through programming, utilizing their mathematical knowledge alongside newly acquired computational thinking skills. This dual focus reinforces the idea that computational thinking can act as a bridge between abstract mathematical concepts and their practical applications (Rodríguez-Martínez et al., 2020; Iskrenovic-Momcilovic, 2020; Molina-Ayuso et al., 2023; Yunianto et al., 2023). Furthermore, the challenges encountered by students, such as difficulty selecting the appropriate Scratch blocks and calculating angles for shapes, underscore the importance of guided practice and scaffolded learning experiences in developing proficiency in computational thinking (Fagerlund et al., 2020; Cui & Ng, 2021).

The accommodation of language and instructional resources to conform to the indigenous context, as manifested through the shift towards simplified Japanese terminology, underscores the profound impact of cultural and linguistic variables on the educational process. Conversely, spatial orientation and visualization serve as pivotal components in shaping students' spatial ability, thereby enhancing their efficacy in mastering mathematical concepts, notably in the domain of geometry (Hendroanto et al., 2018). This adjustment likely contributed to improved student outcomes by reducing cognitive load and enhancing the accessibility of instructions (Bagea, 2023). These findings emphasize the importance of culturally responsive teaching practices in global educational endeavors, particularly in subjects with universal relevance, such as mathematics and computer science. The disparities in performance observed among students in Japan and Indonesia across the three levels of computational thinking skills—Decomposition, Pattern Recognition, Abstraction, and Algorithmic Design—suggest that age, prior exposure to technology, and educational contexts may influence learning outcomes. For instance, the higher proficiency levels demonstrated by Japanese students could be attributed to their earlier and more frequent exposure to digital tools and programming concepts, as proposed by Ogegbo and Ramnarain (2022). Conversely, the notable improvement observed in Indonesian students, particularly at Level 3, may reflect a more significant growth potential stemming from less prior exposure, aligning with findings from Yadav (2017) regarding the impact of introducing computational thinking in less technologically saturated education systems.

Furthermore, as emphasized in previous studies, the significance of integrating Scratch (Sáez-López et al., 2016; Marcelino et al., 2018) and GeoGebra (Yunianto et al., 2023) into school education is apparent to foster students' computational thinking skills in mathematics lessons. Thus far, there has been limited practice in conducting identical classes in Indonesia and Japan and assessing the efficacy of these classes and developmental tools. However, the suggestion made by this research that it may be feasible to enhance the computational thinking skills of students in both countries represents a notable advancement. A forthcoming challenge lies in conducting research on classroom practices in nations beyond Indonesia and Japan. By accumulating teaching practices across a broader spectrum of countries, including developed and developing nations and those where English is not the primary language, we can cultivate human resources poised to shape the future of international society and glean valuable insights into teaching methodologies for their educators. Additionally, while this practice targets elementary school children, conducting lessons for junior high and high school students could offer insights into the characteristics and developmental trajectory of students' computational thinking, their comprehension of polygons, and the learning outcomes associated with Scratch.

CONCLUSION

This study investigated the efficacy of educational practices that foster students' computational thinking skills and comprehension in solving geometry problems from the perspectives of Indonesia and Japan's primary school students. Three key elements were identified as effective in promoting these cognitive abilities by analyzing three classroom sessions. Firstly, aiding children in understanding the concepts inherent in computational thinking was crucial. This was achieved through hands-on activities, visual comparisons of coding blocks, and practical exercises using Scratch programming, which facilitated comprehension of abstract notions like sequential processing and conditional branching. Secondly, integrating mathematical knowledge into the lessons proved beneficial. By applying principles of mathematics, such as those related to polygons, to programming tasks, children gained a deeper understanding of mathematical concepts through their application in different contexts. Lastly, incorporating diverse problem-solving approaches was found to be essential. Activities involving level-based drawing tasks allowed for the utilization of various programming methods, with students demonstrating different problem-solving strategies, including trial and error, pre-calculation, and the use of mathematical tools. Notably, even children facing difficulties in mathematics showed heightened engagement, possibly due to the autonomy afforded by Scratch programming, enabling them to select problem-solving approaches that suited their learning styles. This freedom of choice likely stimulated algorithmic thinking and contributed to the overall effectiveness of the activities in promoting computational thinking skills.

The students in this study not only learned valuable problem-solving skills but also gained broader life lessons, realizing the versatility of problem-solving approaches and language. However, it's important to note that the study was conducted in a single public and private school in both Indonesia and Japan, limiting the generalizability of the findings. Moving forward, a key challenge lies in exploring the adaptability of the classroom practices observed in this study to diverse educational settings. Our future endeavors will focus on accumulating practical research and empirical evidence to further refine and expand upon effective instructional strategies to enhance computational thinking skills through Scratch programming. Recognizing computational thinking as a vital skill for navigating complex challenges, our aim is to cultivate students who are adept at adapting to evolving societal demands. Thus, the continuous development and dissemination of practical research remain crucial for advancing educational practices and fostering the next generation of agile problem-solvers.

Acknowledgments

The researchers wish to express their sincere gratitude for the invaluable collaborative efforts that occurred between two prominent academic institutions during this study. Special recognition is extended to Universitas Ahmad Dahlan in Yogyakarta, Indonesia, and Naruto University of Education in Tokushima, Japan. These esteemed institutions generously provided essential resources, including conducive research environments, expert guidance, and state-of-the-art facilities, which were crucial in facilitating the successful execution of this research endeavor. Additionally, gratitude is extended to one of the public primary schools in Ibaraki Prefecture, Japan, and one of the private schools in Yogyakarta, Indonesia, for granting permission to conduct this research. Finally, special thanks are due to Prof. Ishizaka Hiroki, Taiyu Shigematsu, Isaias González, Tota Endo, Toshio Minamino, Eka Kevin Alghiffari, Sri Rahayu Alam, and Tutik Shahidayanti for their support in this research activity.

Declarations

Author Contribution	:	RCIP: Conceptualization, Supervision, Validation, Writing - Original Draft, and Writing - Review & Editing.
		SK: Conceptualization, Methodology, and Writing - Original Draft.
		NRNP: Investigation, Project Administration, and Writing - Review & Editing.
		HE: Investigation, Data Curation, and Resources.
		AZ: Investigation, Formal Analysis, and Methodology.
		KT: Data Curation, Investigation, and Resources.

Funding Statement : This research was funded by the Institute of Research and Community Service, Universitas Ahmad Dahlan, through the International Research Collaboration Scheme with Fundamental Research Type (Grant Number: 24/RIA/LPPM-UAD/VI/2023) and Naruto University of Education Research Collaboration.

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

Additional Information : Additional information is available for this paper.

REFERENCES

Aminah, N., Sukestiyarno, Y. L., Cahyono, A. N., & Maat, S. M. (2023). Student activities in solving mathematics problems with a computational thinking using Scratch. *International Journal of Evaluation and Research in Education (IJERE),* *12*(2), 613-621. https://doi.org/10.11591/ijere.v12i2.23308

Ardito, G., Czerkawski, B., & Scollins, L. (2020). Learning computational thinking together: Effects of gender differences in collaborative middle school robotics program. *TechTrends, 64,* 373-387. https://doi.org/10.1007/s11528-019-00461-8

Ausiku, M. M., & Matthee, M. C. (2023). A framework for teaching computational thinking in primary schools: A Namibian case study. *The African Journal of Information Systems, 15*(3), 2. https://digitalcommons.kennesaw.edu/cgi/viewcontent.cgi?article=2309&context=ajis

Bagea, I. (2023). Cultural influences in language learning in a global context. *Indo-MathEdu Intellectuals Journal, 4*(2), 630–645. https://doi.org/10.54373/imeij.v4i2.248

Basso, D., Fronza, I., Colombi, A., & Pahl, C. (2018). Improving assessment of computational thinking through a comprehensive framework. *Proceedings of the 18th Koli Calling International Conference on Computing Education Research,* 1-5. https://doi.org/10.1145/3279720.3279735

Bender, J., Zhao, B., Dziena, A., & Kaiser, G. (2023). Integrating Parsons puzzles within Scratch enables efficient computational thinking learning. *Research and Practice in Technology Enhanced Learning, 18*, 022. https://doi.org/10.58459/rptel.2023.18022

Colorafi, K., & Evans, B. (2016). Qualitative descriptive methods in health science research. *HERD: Health Environments Research & Design Journal,* *9*, 16-25. https://doi.org/10.1177/1937586715614171

Cui, Z., & Ng, O. (2021). The interplay between mathematical and computational thinking in primary school students' mathematical problem-solving within a programming environment. *Journal of Educational Computing Research, 59*, 988-1012. https://doi.org/10.1177/0735633120979930

Dahlman, C. (2007). Technology, globalization, and international competitiveness: Challenges for developing countries. *Industrial development for the 21st century: Sustainable development perspectives* (pp. 29-83). Department of Economic and Social Affairs, United Nations. https://sustainabledevelopment.un.org/content/documents/full_report.pdf#page=37

Fagerlund, J., Häkkinen, P., Vesisenaho, M., & Viiri, J. (2020). Assessing 4th grade students' computational thinking through scratch programming projects. *Informatics in Education, 19*(4), 611-640. https://doi.org/10.15388/infedu.2020.27

Global Education Monitoring Report Team - SEAMEO Regional Open Learning Center. (2023). *Technology in education: A case study on Indonesia*. UNESCO https://doi.org/10.54676/WJMY7427

Gökçe, S., & Yenmez, A. A. (2023). Ingenuity of scratch programming on reflective thinking towards problem solving and computational thinking. *Education and Information Technologies, 28*(5), 5493-5517. https://doi.org/10.1007/s10639-022-11385-x

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43. https://doi.org/10.3102/0013189X12463051

Hendroanto, A., van Galen, F., Van Eerde, D., Prahmana, R. C. I., Setyawan, F., & Istiandaru, A. (2018). Photography activities for developing students' spatial orientation and spatial visualization. *Journal of Physics: Conference Series, 943*(1), 012029. http://dx.doi.org/10.1088/1742-6596/943/1/012029

Iskrenovic-Momcilovic, O. (2020). Improving geometry teaching with Scratch. *International Electronic Journal of Mathematics Education, 15*(2), em0582. https://doi.org/10.29333/iejme/7807

Israel-Fishelson, R., & Hershkovitz, A. (2022). Studying interrelations of computational thinking and creativity: A scoping review (2011–2020). *Computers & Education, 176*, 104353. https://doi.org/10.1016/j.compedu.2021.104353

Jiang, B., & Li, Z. (2021). Effect of Scratch on computational thinking skills of Chinese primary school students. *Journal of Computers in Education, 8*(4), 505-525. https://doi.org/10.1007/s40692-021-00190-z

Kale, U., Akcaoglu, M., Cullen, T., Goh, D., Devine, L., Calvert, N., & Grise, K. (2018). Computational what? Relating computational thinking to teaching. *TechTrends, 62*, 574-584. https://doi.org/10.1007/s11528-018-0290-9

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing, 4*(3), 583-596. https://www.bjmc.lu.lv/fileadmin/user_upload/lu_portal/projekti/bjmc/Contents/4_3_15_Kalelioglu.pdf

Kobayashi, Y., & Hasegawa, S. (2020). A proposal for the learning environment to evaluate a computational thinking for elementary school in Japan from operation point of view. *IEICE Technical Report, 120*(167), 13-18.

Kobayashi, Y., Ota, K., & Hasegawa, S. (2022). Development of learning environment to evaluate computational thinking for elementary school students from operation perspective. *Transactions of Japanese Society for Information and Systems in Education, 39*(2), 210-223. https://doi.org/10.14926/jsise.39.210

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning computational thinking and scratch at distance. *Computers in Human Behavior, 80*, 470-477. https://doi.org/10.1016/j.chb.2017.09.025

Molina-Ayuso, Á., Adamuz-Povedano, N., Bracho-López, R., & Torralbo-Rodríguez, M. (2023). Computational thinking with Scratch: A tool to work on geometry in the fifth grade of primary education. *Sustainability, 16*(1), 110. https://doi.org/10.3390/su16010110

Molina-Ayuso, Á., Adamuz-Povedano, N., Bracho-López, R., & Torralbo-Rodríguez, M. (2022). Introduction to computational thinking with Scratch for teacher training for Spanish primary school teachers in mathematics. *Education Sciences, 12*(12), 899. https://doi.org/10.3390/educsci12120899

Ogegbo, A. A., & Ramnarain, U. (2022). Teachers' perceptions of and concerns about integrating computational thinking into science teaching after a professional development activity. *African Journal of Research in Mathematics, Science and Technology Education, 26*(3), 181-191. https://doi.org/10.1080/18117295.2022.2133739

Ono, H., & Zavodny, M. (2007). Digital inequality: A five country comparison using microdata. *Social Science Research, 36*(3), 1135-1155. https://doi.org/10.1016/j.ssresearch.2006.09.001

Piedade, J., & Dorotea, N. (2022). Effects of Scratch-based activities on 4th-grade students' computational thinking skills. *Informatics in Education, 22*(3), 499–523. https://doi.org/10.15388/infedu.2023.19

Purwasih, R., Turmudi, & Dahlan, J. A. (2024). How do you solve number pattern problems through mathematical semiotics analysis and computational thinking?. *Journal on Mathematics Education, 15*(2), 403-430. http://doi.org/10.22342/jme.v15i2.pp403-430

Rafiepour, A., & Farsani, D. (2021). Cultural historical analysis of Iranian school mathematics curriculum: The role of computational thinking. *Journal on Mathematics Education, 12*(3), 411-426. http://doi.org/10.22342/jme.12.3.14296.411-426

Robinson, L., Cotten, S. R., Ono, H., Quan-Haase, A., Mesch, G., Chen, W., ... & Stern, M. J. (2015). Digital inequalities and why they matter. *Information, Communication & Society, 18*(5), 569-582. https://doi.org/10.1080/1369118X.2015.1012532

Rodríguez-Martínez, J. A., González-Calero, J. A., & Sáez-López, J. M. (2020). Computational thinking and mathematics using Scratch: An experiment with sixth-grade students. *Interactive Learning Environments, 28*(3), 316-327. https://doi.org/10.1080/10494820.2019.1612448

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. *Computational Thinking Education*, 79-98. Springer. https://doi.org/10.1007/978-981-13-6528-7

Roque, R., Rusk, N., & Resnick, M. (2016). Supporting diverse and creative collaboration in the Scratch online community. In Cress, U., Moskaliuk, J., Jeong, H. (Eds.) *Mass Collaboration and Education. Computer-Supported Collaborative Learning Series*, vol 16 (pp. 241–256). Springer. https://doi.org/10.1007/978-3-319-13536-6_12

Rose, S. (2019). Developing children's computational thinking using programming games. *Doctoral Thesis*. Sheffield Hallam University. https://doi.org/10.7190/shu-thesis-00257

Sáez-López, J., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computer & Education, 97*, 129-141. https://doi.org/10.1016/j.compedu.2016.03.003

Saritepeci, M. (2020). Developing computational thinking skills of high school students: Design-based learning activities and programming tasks. *The Asia-Pacific Education Researcher, 29*(1), 35-54. https://doi.org/10.1007/s40299-019-00480-2

Shute, V., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, *22*, 142-158. https://doi.org/10.1016/J.EDUREV.2017.09.003

Smith, S. M., Novak, E., Schenker, J., & Kuo, C. L. (2022). Effects of computer-based (Scratch) and robotic (Cozmo) coding instruction on seventh grade students' computational thinking, competency beliefs, and engagement. In Kim, JH., Singh, M., Khan, J., Tiwary, U.S., Sur, M., Singh, D. (Eds.) *Intelligent Human Computer Interaction. IHCI 2021. Lecture Notes in Computer Science*, vol 13184 (pp. 325–336). Springer. https://doi.org/10.1007/978-3-030-98404-5_31

Sparrow, R., Dartanto, T., & Hartwig, R. (2020). Indonesia under the new normal: Challenges and the way ahead. *Bulletin of Indonesian Economic Studies, 56*(3), 269-299. https://doi.org/10.1080/00074918.2020.1854079

Stewart, W., & Baek, K. (2023). Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. *International Journal of Computer Science Education in Schools, 6*(1), 35-58. https://doi.org/10.21585/ijcses.v6i1.156

Willis, D., Sullivan-Bolyai, S., Knafl, K., & Cohen, M. (2016). Distinguishing features and similarities between descriptive phenomenological and qualitative description research. *Western Journal of Nursing Research, 38*, 1185-1204. https://doi.org/10.1177/0193945916645499

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35. https://doi.org/10.1145/1118178.1118215

Yadav, A., Good, J., Voogt, J., & Fisser, P. (2017). Computational thinking as an emerging competence domain. In Mulder, M. (Ed.) *Competence-based Vocational and Professional Education. Technical and Vocational Education and Training: Issues, Concerns and Prospects*, vol 23 (pp. 1051–1067). Springer. https://doi.org/10.1007/978-3-319-41713-4_49

Yunianto, W., Bautista, G., Prahmana, R. C. I., & Lavicza, Z. (2023). GeoGebra applet to learn programming and debugging in mathematics lessons. *Proceedings of 2023 International Symposium on Computers in Education (SIIE)* (pp. 1-5). IEEE. https://doi.org/10.1109/SIIE59826.2023.10423707

Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education, 141*, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Zhao, L., Liu, X., Wang, C., & Su, Y. S. (2022). Effect of different mind mapping approaches on primary school students' computational thinking skills during visual programming learning. *Computers & Education, 181*, 104445. https://doi.org/10.1016/j.compedu.2022.104445