

Improving Grade 7 Learners' Problem-Solving Skills Through Collaborative Coding in Scratch Programming with the Use of Chromebooks

Sabangan, Esperanza R.

Department of Education, Makati City, National Capital Region, Philippines

DOI: <https://dx.doi.org/10.47772/IJRISS.2025.903SEDU0550>

Received: 14 September 2025; Accepted: 20 September 2025; Published: 17 October 2025

ABSTRACT

This study investigated whether Chromebooks and Scratch collaborative coding could improve Grade 7 problem-solving skills. During the organized intervention, students completed coding homework in groups. The program taught them reasoning, critical thinking, and teamwork. Results demonstrated that learners improve at problem-solving, code-fixing, and pattern-spotting. They scored 89.30% on computational thinking tests. Students gained confidence and were more open to new ideas by studying together. Survey findings showed that individuals were involved, and Chromebooks made real-time collaboration and feedback easier. According to the study, structured group coding, guided reflection, and peer mentorship improved students' problem-solving skills and closed performance discrepancies.

The study suggested gamification strategies such as debugging races, scaffolded collaborative projects, and Scratch-based teacher training. Future research should examine how people recall abilities and how to apply collaborative coding in many fields. Scratch with cooperation on Chromebooks enhances problem-solving, creativity, and computational thinking. This approach makes middle school ICT instruction effective.

ACKNOWLEDGMENT

The Researcher expresses her profound gratitude to the Almighty God, the source of wisdom and power, for His grace throughout this study.

The passion of my grade 7 students for Scratch programming made this research worthwhile. This study began because they wanted to share their PBL experiences. My colleagues and school management deserve praise for encouraging creative teaching and a welcoming environment.

The ICT experts provided technical assistance and constructive feedback throughout the research process. They also served as study reviewers and validated the survey questionnaire.

Dr. Renald Jay O. Fio, Philippine Merchant Marine School, Inc., for his valuable feedback; Dr. Venus M. Mariano, a master teacher at Makati High School, validated the survey questionnaire.

Rowena A. Reyes, MSIT, Senior Education Program Specialist—SMN Schools Division Office-Makati, Department of Education, provided assistance in extending her expertise in information technology.

Finally, I appreciate my family's patience and support during research and writing. I am grateful to everyone who contributed to the success of this study.

Context And Rationale

Developing problem-solving skills in programming was a critical component of ICT education, particularly for Grade 7 learners engaged in Scratch programming. However, the first-quarter summative skills assessment data from the researcher's Grade 7 ICT classes showed that students struggled significantly in this area. The assessment revealed an overall mean percentage score (MPS) of 71.06, with a mastery level of only 8.83%.

This figure translated to just 27.32% (53 out of 194 learners) achieving a passing score of 75% or higher, implying that nearly 70% of the students faced challenges in problem-solving, creativity, and logical thinking. These findings were particularly concerning given the importance of these skills in Scratch programming, where students had to develop and debug complex coding sequences. The data aligned with the researcher's classroom observations, where many students demonstrated a reliance on trial-and-error approaches and struggled to apply coding concepts independently. This difficulty suggested a need for instructional strategies that better supported problem-solving development, particularly in collaborative settings where learners could share insights and strategies.

Recent studies have highlighted the problem-solving skills of Filipino learners and reinforced the urgency of addressing these deficits. The Southeast Asian Ministers of Education Organization (SEAMEO) conducted a national assessment, which revealed that Filipino students, especially in STEM-related subjects, lagged behind their peers in neighboring countries in problem-solving competencies (SEAMEO, 2021). In the context of ICT education, where computational thinking was key, these gaps hindered not only academic performance but also the future readiness of students for more advanced technology courses. Failure to address this issue in middle school could have limited students' ability to engage in critical 21st-century skills, including coding, innovation, and digital literacy, which are increasingly essential in the global workforce.

Research further supported the notion that collaboration could significantly enhance problem-solving abilities. Studies by Maligalig and Tan (2020) indicated that students who worked in collaborative coding environments developed stronger critical thinking skills and were more adept at troubleshooting programming issues than those who worked alone. This finding was particularly relevant in the context of Scratch, where learners could benefit from peer interaction to solve coding puzzles, share different approaches to problem-solving, and gain confidence in their coding abilities. Collaborative coding helped alleviate the frustration that came from working in isolation and encouraged persistence through peer support.

The integration of Chromebooks into this process was a crucial factor in enhancing collaboration. A study by Dela Cruz (2022) highlighted that Chromebooks, when used in classrooms, promoted real-time collaboration, allowing students to co-create and share projects seamlessly. This research proposed that by using Chromebooks as part of a collaborative Scratch programming environment, students could engage in meaningful group problem-solving activities, boosting their confidence and ability to apply coding concepts. Therefore, conducting this action research was essential to explore how collaborative coding with Chromebooks in Scratch programming could bridge the problem-solving skill gap and equip Grade 7 learners with the necessary tools to succeed in both academic and future professional settings.

This action research aimed to investigate the impact of collaborative coding in Scratch programming using Chromebooks on the problem-solving skills of Grade 7 learners. By fostering teamwork and peer support, the study sought to create a learning environment where students could collaboratively approach coding tasks, thereby improving their ability to think critically, solve problems logically, and apply programming concepts more effectively. Addressing these gaps was vital in equipping students with the skills needed for future technological challenges and ensuring their success in ICT education and beyond.

Action Research Questions

This study investigated how collaborative coding in Scratch programming using Chromebooks enhanced Grade 7 learners' problem-solving skills. The research explored how teamwork in coding projects influenced students' critical and creative thinking when addressing programming challenges. It examined collaborative coding strategies, the role of Chromebooks in facilitating interaction and communication, and whether this approach increased learners' confidence and engagement in coding. The study revealed how technology-enhanced collaborative learning improved ICT problem-solving skills and addressed the following research questions:

1. What is the level of problem-solving skills of the Grade 7 learners in Scratch programming?
2. What is the level of problem-solving strategies in Scratch programming of the Grade 7 learners?
3. What is the combined level of critical thinking and creativity of the Grade 7 learners?

4. To what extent do Chromebooks facilitate collaborative learning among Grade 7 learners?
5. To what extent do teamwork and communication contribute to the learners' collaborative experiences in Scratch programming?
6. How effective is collaborative learning with Chromebooks in enhancing learners' problem-solving, creativity, and critical thinking in Scratch programming?
7. What is the performance of the students in Scratch Programming based on their 3rd quarter assessment?
8. Is there a relationship between the effectiveness of collaborative learning with Chromebooks in enhancing learners' problem-solving creativity and critical thinking and their 3rd quarter assessment in Scratch programming?
9. What additional strategies or activities can be implemented to further enhance Grade 7 learners' problem-solving skills in Scratch programming beyond the use of Chromebooks and collaborative learning?

Hypothesis

The formulated hypothesis was tested for fulfillment in this study.

There is no significant relationship between the effectiveness of collaborative learning with Chromebooks in enhancing learners' problem-solving creativity and critical thinking and their third-quarter assessment in Scratch programming.

Innovation, Intervention, And Strategy

The innovation integrated collaborative coding in Scratch programming with the use of Chromebooks to improve problem-solving skills among Grade 7 learners. Based on the researcher's classroom observations, students often struggled with coding tasks when working independently. However, through structured collaboration, learners shared ideas, troubleshooted together, and developed critical thinking skills in a supportive environment. The use of Chromebooks enabled real-time collaboration and enhanced peer-to-peer interaction, facilitating a more interactive and engaging learning experience. As noted by Pierson and Williams (2022), technology-mediated collaboration fostered deeper learning and engagement in problem-solving tasks. This intervention aimed to create a learning environment where students worked together to overcome coding challenges, thereby strengthening their problem-solving abilities through active collaboration and hands-on practice (Zhong & Song, 2023).

The primary idea behind this action research was to enhance Grade 7 learners' problem-solving skills through collaborative coding in Scratch programming using Chromebooks. The researcher observed that when students worked together on coding tasks, they became more engaged and actively participated in problem-solving. This approach leveraged the benefits of collaborative learning, where students could share ideas, debug together, and develop computational thinking in a more interactive environment (Grover & Pea, 2021). The use of Chromebooks facilitated real-time collaboration and allowed for seamless project sharing and peer feedback.

This research addressed the issue of many Grade 7 learners struggling with independent coding tasks, particularly when faced with complex programming challenges. The researcher found that students often became frustrated or disengaged when they could not resolve coding errors on their own. This difficulty in solving programming challenges independently had been a barrier to developing their problem-solving skills and confidence (Zhong & Song, 2023). The lack of peer collaboration and support in traditional coding setups may have limited opportunities for students to learn from one another.

The proposed system involved implementing collaborative coding sessions using Chromebooks in Scratch programming lessons. During these sessions, the researcher grouped students and assigned them coding challenges that necessitated teamwork, idea sharing, and mutual problem-solving. The study used Chromebooks as tools to facilitate real-time collaboration, with students working together on shared Scratch projects through cloud-based applications. This system promoted both active learning and the use of technology to foster interaction and communication during problem-solving activities (Pierson & Williams, 2022).

This research proposed the integration of collaborative coding activities, where students participated in paired or group programming using Chromebooks. The researcher expected learners to enhance their coding problem-solving skills through collaborative discussion, peer feedback, and group brainstorming. This collaborative approach enabled students to develop a more profound understanding of coding concepts, enhanced their critical thinking skills, and built confidence in solving programming tasks (Grover & Pea, 2021).

The researcher observed an increase in student engagement, confidence, and problem-solving abilities when they collaborated on coding projects using Chromebooks. Learners who collaborated tended to be more motivated and persistent in overcoming challenges. Research had shown that technology-enhanced collaborative learning fostered more profound understanding and active participation in problem-solving tasks (Zhong & Song, 2023). Through this action research, the researcher sought to explore how these dynamics unfolded in the context of Scratch programming with Grade 7 students.

To strengthen the problem-solving skills of Grade 7 learners, educators utilized a combination of collaborative learning, Scratch programming, and technology integration through Chromebooks. Collaborative learning, particularly in coding, encouraged students to work together to solve programming challenges. Scratch, a block-based visual programming language, served as an accessible platform for beginners to engage in computational thinking. Chromebooks, as lightweight and versatile devices, allowed students to work in real-time on shared coding projects, enhancing both collaboration and immediate feedback. Studies by Pierson and Williams (2022) highlighted the effectiveness of combining technology tools with collaborative learning to foster problem-solving and critical thinking in educational settings. Additionally, integrating Chromebooks supported continuous learning beyond the classroom, as students could access their Scratch projects from anywhere, encouraging more engagement with the task (Zhong & Song, 2023).

The researcher's classroom experiences and previous research served as the foundation for the proposed intervention of collaborative coding using Chromebooks. Research had consistently shown that collaborative coding fostered problem-solving by allowing students to exchange ideas, build upon each other's strengths, and develop solutions collectively (Grover & Pea, 2021). In the researcher's teaching practice, students who worked together in coding activities were often more successful at completing complex tasks compared to those who worked independently. By using Scratch, a user-friendly programming environment, students were able to focus more on problem-solving strategies and less on syntax, thus increasing their engagement. Chromebooks, as digital tools, provided the flexibility needed for seamless collaboration and access to learning resources. Earlier research by Wastiau and Rodríguez (2021) supported the idea that using technology for group coding activities greatly improved how well students learned.

The rationale behind using collaborative coding in Scratch with Chromebooks was to address the lack of problem-solving skills observed when students worked independently on coding tasks. Collaboration allowed learners to leverage peer support, share diverse perspectives, and collectively debug problems, which improved their problem-solving capabilities. The extent of the intervention included structured collaborative coding sessions in class, where students worked in groups on assigned tasks, utilizing Chromebooks for real-time collaboration. However, the strategy had its limitations. Some students relied too heavily on their peers, potentially diminishing individual accountability. Additionally, access to Chromebooks outside of class was a limitation for students with limited technology access at home. Despite these limitations, the strategy was plausible as a solution because it directly addressed the identified problem by providing structured opportunities for learners to engage with each other and solve programming challenges together (Wastiau & Rodríguez, 2021). By integrating technology with collaborative learning, this approach aligned with the constructivist model of learning, where students built knowledge through shared experiences and peer interaction (Pierson & Williams, 2022).

ACTION RESEARCH METHODS

The action research methods employed in this study aimed to systematically investigate the impact of collaborative coding in Scratch programming on Grade 7 learners' problem-solving skills while utilizing Chromebooks. Drawing on the researcher's personal experiences, the approach involved a mixed-methods design that combined quantitative and qualitative data collection. The researcher collected numerical data from

surveys and tests to see how students' problem-solving skills changed before and after the program and also gathered descriptive data from student reflections, group talks, and notes to understand how they worked together during coding sessions. This dual approach aligned with recommendations from Mertler (2020), who emphasized the importance of using both quantitative and qualitative methods to develop an in-depth understanding of educational practices. The study also used surveys to gauge students' engagement and confidence with programming challenges. This conclusion was similar to what other studies had found, which was that collaborative learning environments boosted student motivation and self-efficacy (Hwang et al., 2021). This action research method not only aimed to improve learners' problem-solving skills but also fostered a supportive and interactive learning environment that encouraged collaboration among students.

A. Participants and Other Sources of Data and Information

In this action research study, the participants included a diverse group of 33 Grade 7 students enrolled in a Scratch programming course. To provide a comprehensive understanding of how collaborative coding influenced problem-solving skills across different learner profiles, the study selected participants who represented various skill levels and backgrounds. The students engaged in structured collaborative coding activities using Chromebooks, which allowed for real-time collaboration and peer support during the coding process. Additionally, the study included insights from teachers who had facilitated similar coding experiences, providing a broader context for the findings. The researcher interviewed these educators to gather qualitative data on their observations of student engagement, collaboration, and problem-solving behaviors in the classroom.

Data collection methods included different sources, such as tests before and after the intervention to measure changes in problem-solving skills and surveys to understand how students felt about working together on coding and using technology, and journals where students wrote about their experiences and what they learned. This multifaceted approach aligned with the recommendations of Creswell (2021), who emphasized the value of triangulating data sources to enrich the findings and provide a holistic view of the research topic. Additionally, the researcher took notes during group coding sessions, allowing for immediate examination of how students interacted, communicated, and solved problems. This combination of quantitative and qualitative data sources ensured a robust analysis of the impact of collaborative coding on Grade 7 learners' problem-solving skills.

B. Methods

To effectively assess the impact of collaborative coding in Scratch programming on Grade 7 learners' problem-solving skills, this action research utilized various instruments designed to capture both quantitative and qualitative data. The researcher administered pre- and post-tests to assess the students' problem-solving abilities before and after the intervention, enabling a measurable comparison of skill development. The researcher developed these tests based on Bloom's Taxonomy, which encompassed various cognitive levels from basic recall to higher-order thinking skills (Anderson & Krathwohl, 2001). Structured surveys measured student engagement and confidence by incorporating Likert-scale questions to gauge students' perceptions of their collaboration experiences and self-efficacy in coding tasks. The study also implemented reflective journals, where students documented their learning experiences, challenges encountered, and strategies developed during the coding sessions. This mixed-methods approach provided a comprehensive understanding of how collaborative coding influenced learners' problem-solving skills and engagement levels, as supported by Hwang et al. (2021), who emphasized the effectiveness of varied assessment tools in educational research.

This research study systematically organized the data collection procedures to ensure the efficient and effective gathering of relevant information. The researcher administered pretests initially to establish baseline problem-solving skills among the participants. Following this, students engaged in a series of collaborative coding activities using Chromebooks over a period of four weeks. The researcher conducted observations during these sessions to document the students' interactions, communication patterns, and problem-solving strategies, thereby offering real-time insights into collaborative processes. The researcher administered post-tests after the intervention to assess the changes in students' problem-solving abilities. At the end of the coding sessions, the researcher distributed surveys to assess students' attitudes toward collaboration and technology integration in

their learning. Finally, the researcher qualitatively analyzed the reflective journals to identify common themes related to the students' experiences and perceptions of their learning journey. Creswell (2021) recommended best practices that aligned with this structured data collection process, thereby enhancing the reliability and validity of the research findings.

C. Data Analysis Plan

The data analysis for this action research study employed a mixed-methods approach, integrating both quantitative and qualitative data to provide a comprehensive view of how collaborative coding in Scratch programming affected Grade 7 learners' problem-solving skills. The researcher looked at the numbers from tests taken before and after the program using simple statistics, like averages and differences, to see if there were important changes in problem-solving skills before and after the program (Field, 2022). This statistical analysis highlighted the effectiveness of the collaborative coding strategy in enhancing students' skills. Furthermore, the researcher visually represented the results through charts and graphs to enhance understanding and communication of the findings. The study organized and analyzed information from students' reflective journals and observations to find common themes and insights about how they worked together, participated, and used strategies during coding tasks. Thematic analysis allowed for a deeper appreciation for student experiences and perceptions, as recommended by Braun and Clarke (2021). The study integrated the findings from both data types to provide a comprehensive understanding of the intervention's impact. The researcher reported these findings in a structured format, combining statistical results with illustrative qualitative quotes and examples.

The analysis of data utilized both qualitative and quantitative methods to ensure a robust interpretation of the research findings. Quantitative analysis focused on the numerical data gathered from the pre- and post-tests and surveys, utilizing statistical software (such as SPSS or Excel) for calculations. This analysis provided objective measures of student performance improvements in problem-solving skills and engagement levels. The study employed thematic analysis on the qualitative side to examine reflective journal entries and observational notes. This process involved coding the data into meaningful themes that reflected the participants' collaborative experiences and the problem-solving strategies they developed. The qualitative insights complemented the quantitative data, enriching the overall understanding of the intervention's effectiveness. Using a mix of methods, this study examined the complicated learning processes involved in collaborative coding. The result was in line with the findings of Hwang et al. (2021), which stressed how important it was to use a variety of analytical methods in educational research to fully understand how students learn in all its forms.

Statistical Treatment

The study, entitled "*Improving Grade 7 Learners' Problem-Solving Skills Through Collaborative Coding in Scratch Programming with the Use of Chromebooks*," examined how Chromebooks affected students' performance in Scratch programming.

Weighted Mean (Team, 2024)

It is a statistical approach for calculating the average by multiplying the weights with their respective means and taking their sum. This is used to summarize the response regarding the perception or experience of students in terms of access, affordability, and impact on their academic and social life.

Formula:

$$W = \frac{\sum_{i=1}^n w_i X_i}{\sum_{i=1}^n w_i}$$

Where:

W = Weighted Average

n = Number of terms to be averaged

w = Weights applied to x values

X = Data values to be average

Pearson Product Moment Coefficient of Correlation (Pearson's r)

Pearson's r was used to measure the strength and direction of the linear relationship between two variables, serving as a key statistical tool in the analysis.

A positive correlation (r close to +1) between collaborative learning efficacy and assessment scores supported the hypothesis that Chromebook-based teamwork enhanced problem-solving in Scratch programming. Conversely, a weak or negative correlation suggested the need to explore alternative instructional strategies.

This quantitative study assessed whether collaborative coding on Chromebooks improved Grade 7 students' computational thinking and problem-solving abilities.

DISCUSSION OF RESULTS AND REFLECTION

This section presents the discussion of interpretation and analysis of data gathered from the study on how collaborative coding in Scratch using Chromebooks positively influenced Grade 7 learners' problem-solving skills. Here's the detailed discussion of the results of the study.

The Levels of Problem-Solving Skills in Scratch Programming of the Grade 7 Learners in Scratch Programming

Table 1 Level of Problem-Solving Skills in Scratch Programming

Problem Solving Skills in Scratch Programming	Weighted Mean	Verbal Interpretation
1. Collaborative coding in Scratch has improved my ability to solve coding challenges effectively.	3.71	Very High
2. Working with my classmates during Scratch projects helps me think of new and better ways to solve problems.	3.63	Very High
3. Collaborative coding tasks have taught me how to debug errors in Scratch programs more efficiently.	3.65	Very High
4. I feel more confident in solving complex programming tasks after participating in collaborative Scratch activities.	3.72	Very High
5. Collaborative coding has helped me understand how to approach problem-solving step by step in Scratch.	3.65	Very High
Overall Mean	3.67	Very High

Table 1 displays the level of problem-solving skills among students in Scratch programming, using a weighted mean and its corresponding verbal interpretation. The overall mean of 3.67 showed that Grade 7 students have excellent Scratch programming problem-solving skills. Students significantly agreed that collaborative coding improved their ability to solve coding tasks (WM = 3.71), suggesting that teamwork improved their computational thinking. This supports Zhang et al.'s (2021) findings that collaborative learning in block-based programming environments improves critical thinking and problem-solving. As Kong & Wang (2020) stated, peer contact encourages creative problem-solving, and students felt that engaging with classmates helped them create new solutions (WM = 3.63).

High results in debugging efficiency (WM = 3.65) and step-by-step problem-solving (WM = 3.65) showed that learners organized their problem-solving approaches, which are crucial to computational thinking (Grover &

Pea, 2023). The statement with the highest score (WM = 3.72) indicated that learners felt more confident when facing challenging tasks, which supports Atmatzidou & Demetriadis (2022) in their assertion that collaboration in programming enhances problem-solving confidence.

The results show that the students who coded in Scratch together solved problems better, scoring Very High on all indicators. The learners' better debugging, systematic reasoning, and confidence in challenging assignments indicated how collaborative programming improves computational thinking. These results back up what other studies have found about the benefits of peer-assisted learning in STEM education (Hsu et al., 2021).

Triangulation Of Data Analysis

The level of problem-solving skills of the grade 7 learners in Scratch programming to their performance in the 3rd quarter assessment and to their responses to open-ended survey questions.

The study used quantitative tests, performance indicators, and qualitative feedback to evaluate Grade 7 students' problem-solving skills in Scratch programming. The research indicated that Chromebook collaborative coding improved learners' performance.

The quantitative survey results indicated that respondents had high problem-solving skills, notably in debugging (WM = 3.65) and confidence to manage obstacles (WM = 3.72).

The third quarter assessment results indicate that all students scored well, with a high mean score (44.65/50) and MPS (89.30%) and a low standard deviation (SD = 0.2787).

Qualitative responses supported these findings. Students reported that teamwork helped them solve problems and use logic outside coding, supporting Brennan and Resnick (2021)'s findings on computational transfer. To strengthen their problem-solving skills, some students requested peer mentoring and unplugged planning (Bers et al., 2023).

Triangulation Finally, survey, performance, and open-ended response data show that collaborative Scratch programming on Chromebooks improves problem-solving. However, systematic peer mentoring, gamification, and interdisciplinary initiatives could boost these benefits.

The Levels of Problem-Solving Strategies in Scratch Programming of the Grade 7 Learners

Table 2 Level of Problem-Solving Strategies in Scratch Programming

Problem Solving Strategies in Scratch Programming	Weighted Mean	Verbal Interpretation
1. I can break down a big problem into smaller, manageable parts when coding in Scratch.	3.71	Very High
2. I use logical thinking to find and fix errors (debugging) in my Scratch projects.	3.73	Very High
3. I can identify patterns in coding problems and use them to solve similar challenges.	3.79	Very High
4. I feel more confident in solving complex programming tasks after participating in collaborative Scratch activities.	3.76	Very High
5. Problem-Solving Strategies in Scratch When I face a coding problem, I try different solutions before asking for help.	3.71	Very High
Overall Mean	3.74	Very High

Table 2 displays the level of problem-solving strategies among students in Scratch programming, using a weighted mean and its corresponding verbal interpretation. The overall weighted mean of 3.74 demonstrated that Grade 7 students exhibited very high Scratch programming problem-solving skills. Students excelled in seeing patterns (WM = 3.79), solving problems logically (WM = 3.73), and breaking down complex

difficulties (WM = 3.71). Lye & Koh (2020) found that block-based programming environments like Scratch increase computational thinking by encouraging methodical issue decomposition and pattern recognition. Coding together made pupils more confident in difficult tasks (WM = 3.76). This confirms Hsu et al.'s (2021) claim that programming with peers fosters determination and experimentation.

The high mean scores indicated that Scratch's interactive and visible nature helped pupils solve challenges. According to Brennan & Resnick (2020), Scratch teaches kids how to debug and reason iteratively, which are crucial to solving computer problems. Learners' tendency to search for multiple answers before asking for help (WM = 3.71) indicates greater self-confidence. Grover & Pea (2023) observed similar findings on scaffolding in coding instruction.

The results showed that seventh-graders were better at Scratch problem-solving, deconstruction, debugging, and pattern identification. Using Chromebooks together made them more confident and open to new ideas. This suggests that Scratch and peer learning are effective tools for teaching computational problem-solving in early secondary education. These results support the idea that pair programming strategies could be systematically implemented to reinforce collaborative problem-solving, following Denner et al.'s (2021) findings on peer learning efficacy.

Triangulation Of Data Analysis

The level of problem-solving strategies of the grade 7 learners in Scratch programming to their performance in the 3rd quarter assessment and to their responses to open-ended survey questions.

Through collaborative coding on Chromebooks, Grade 7 students learned advanced Scratch problem-solving strategies. Three main data sources show this:

Pattern recognition (WM=3.79), logical problem-solving (WM=3.73), and task decomposition (WM=3.71) were students' strengths. Lye and Koh (2020) found that block-based programming improves reasoning. Collaborative coding increased self-confidence (WM=3.76), supporting Hsu et al.'s (2021) finding that peers help maintain coding commitment.

All learners in the class solve problems well, as shown by the high mean score (44.65 out of 50) and low variability. This supports Zhang et al.'s (2020) claim that real-time collaboration improves computational thinking. The small difference in scores (38–49) indicates that the program helped everyone perform better, backing up Kalelioglu & Gülbahar (2023)'s research on fair learning in group programming.

Students stated peer feedback ("My partner assisted me in identifying errors more swiftly") and transferable tactics ("I now decompose large problems into steps in mathematics as well") strengthened their debugging skills. The Scratch iterative reasoning method used by Brennan and Resnick (2020) matches these qualitative observations. They show that formal scaffolding like unplugged planning activities works (Bers et al., 2023).

The triangulated data analysis indicates that peer learning and observable, iterative design processes in collaborative Scratch programming on Chromebooks improve problem-solving skills. Later phases may incorporate gamified debugging problems (Kafai & Burke, 2024) and transdisciplinary projects (Resnick et al., 2020) to enhance field proficiency.

The Combined Levels of Critical Thinking and Creativity in Scratch Programming of the Grade 7 Learners

Table 3 Combined Level of Critical Thinking and Creativity

Critical Thinking and Creativity	Weighted Mean	Verbal Interpretation
1. I can think of different ways to solve a problem when coding in Scratch.	3.78	Very High
2. I experiment with new coding techniques to create better Scratch projects.	3.78	Very High
3. Scratch programming helps me improve my ability to think logically and critically.	3.87	Very High

4. I can explain my coding solutions clearly to my classmates and teacher.	3.63	Very High
5. I can predict potential errors in my Scratch code before running the program.	3.59	Very High
Overall Mean	3.73	Very High

Table 3 displays the combined level of critical thinking and creativity among students in Scratch programming, using a weighted mean and its corresponding verbal interpretation. The findings showed that when Grade 7 students worked together to code with Scratch programming on Chromebooks, they had a very high level of both critical thinking and creativity. The overall mean of 3.73 (very high) showed that these skills were outstanding. Learners demonstrated excellent logical and critical thinking abilities (WM = 3.87), suggesting that practicing Scratch programming strengthened their problem-solving skills, similar to the findings of Sáez-López et al. (2020), which indicated that block-based programming enhances computational thinking. Furthermore, students showed creative problem-solving (WM = 3.78 for both Q1 and Q2), which means they were open to trying out different answers and new methods. Such behavior is an important part of creativity in coding, as Kong et al. (2022) point out.

The learners got better at analyzing things by being able to explain code solutions (WM = 3.63) and guess what mistakes would happen before they happened (WM = 3.59). These abilities are essential for debugging and iterative design processes (Grover & Pea, 2023). These results showed that collaborating on Scratch coding not only enhanced technical abilities but also increased learners' awareness of their own thinking by allowing them to explain and refine their thought processes.

Collaborative Scratch programming using Chromebooks dramatically boosted grade seven students' creativity and critical thinking. The intervention improved problem-solving adaptability, logical reasoning, and creative exploration, as seen by the high results in all areas. These results support Zhang & Nouri (2021), who found that coding together promotes higher-order thinking. Students' ability to predict issues and explain remedies was applicable beyond programming.

Triangulation Of Data Analysis

The combined level of critical thinking and creativity of the grade 7 learners in Scratch programming to their performance in the 3rd quarter assessment and to their responses to open-ended survey questions.

Chromebooks for collaborative Scratch coding improved seventh-graders' critical thinking and creativity, according to studies. Three datasets agree:

Students showed strong logical reasoning (WM = 3.87) and innovative problem-solving (WM = 3.78), supporting Sáez-López et al.'s (2020) findings on block-based programming and computational thinking. The students' ability to foresee failures (WM = 3.59) and express corrective actions (WM = 3.63) suggests organized debugging skills, consistent with Grover & Pea's (2023) findings on iterative design in programming.

The class's high mean score (44.65/50) and minimal variability show that all pupils comprehended creative-logical integration. This supports Zhang et al. (2020), who found that collaborative real-time involvement promotes varied cognitive methods and analytical rigor. The narrow score range (38–49) supports Kalelioglu & Gülbahar's (2023) claim that cooperative programming is accessible to anyone.

Students reported skills including "decomposing problems into sequential steps for mathematics" and "evaluating multiple solutions prior to selection." These ideas support Kong et al.'s (2022) views on programming creativity and Brennan & Resnick's (2021) metacognition in debugging. Collaboration can inspire innovative ideas, as several students said their teammates' concepts helped them think differently (Hsu et al., 2022).

The triangulated data shows that Scratch programming on Chromebooks promotes critical thinking and creativity through systematic problem-solving and experimental flexibility. Future incarnations could include debugging, flowcharting, and cross-disciplinary projects in Scratch.

The Extent of Chromebooks to Facilitate Collaborative Learning Among Grade 7 Learners

Table 4 Extent to which Chromebooks Facilitate Collaborative Learning

The Role of Chromebooks in Collaborative Learning	Weighted Mean	Verbal Interpretation
1. The use of Chromebooks makes it easier to work together with my classmates on Scratch programming projects.	3.59	To a great extent
2. Chromebooks provide the necessary tools and features for efficient collaborative coding in Scratch.	3.49	To a great extent
3. Using Chromebooks for Scratch programming allows us to share ideas and code more effectively.	3.60	To a great extent
4. Chromebooks make it more convenient to access and edit Scratch projects in real-time with my team.	3.57	To a great extent
5. Technology like Chromebooks enhances my overall experience in collaborative Scratch programming.	3.60	To a great extent
Overall Mean	3.57	To great extent

Table 4 displays the extent to which Chromebooks facilitate collaborative learning among students in Scratch programming, using a weighted mean and its corresponding verbal interpretation. Based on the results, the Grade 7 students found Chromebooks useful for Scratch collaboration. All five items received weighted averages ranging from 3.49 to 3.60, indicating a positive response to the poll. To be more specific, students strongly agreed that Chromebooks made it easier to collaborate (3.59), code efficiently (3.49), share ideas and codes (3.60), access and edit projects in real time (3.57), and improve their overall collaborative experience. The overall mean score of 3.57 demonstrated that Chromebooks helped collaborative learning.

These findings support the current study on how technology might improve collaboration. Hwang and Fu (2020) claim Chromebooks make it easier for students to collaborate and share resources, encouraging interactive learning. Smith et al. (2022) revealed that cloud-based devices make real-time programming collaboration simpler, supporting the students' claims that it was straightforward and accessible. Chromebooks are popular for Scratch programming, which illustrates that they connect technical and social problem-solving.

Chromebooks were crucial for Grade 7 students learning Scratch programming together, according to data. Chromebooks helped students communicate, collaborate, and stay motivated in coding projects, as well as provide tech support. This finding aligns with García-Martínez et al. (2023): using accessible digital tools in programming training improves peer learning and problem-solving skills. Chromebooks made coding collaborative thanks to their high average scores in all survey categories.

Triangulation Of Data Analysis

The extent of Chromebooks in facilitating collaborative learning among grade 7 learners in Scratch programming to their performance in the 3rd quarter assessment and to their responses to open-ended survey questions.

The study found that Chromebooks helped Grade 7 pupils collaborate on Scratch programming. Quantitative, qualitative, and performance-based evidence clearly demonstrated their effectiveness.

The weighted mean ratings, 3.49–3.60, showed that most students thought Chromebooks enabled real-time collaboration, code sharing, and peer contact. Hwang and Fu (2020) found that cloud-based devices like Chromebooks help programmers collaborate. Chromebooks are easy to use and compatible with web-based tools like Scratch, allowing students to focus on problem-solving rather than gadget operation, according to Smith et al. (2022).

The third quarterly evaluation showed that Chromebook cooperation improved students' problem-solving skills (MPS = 89.30%, SD = 0.2787). Organized digital collaboration reduces performance differences, as Grover et al. (2021) found with low score variance.

Students' qualitative comments confirmed that Chromebooks' real-time editing and sharing enabled collaboration and idea sharing. García-Martínez et al. (2023) agree that real-time collaborative technologies improve computational thinking engagement.

The triangulated data shows that Chromebooks help students collaborate on Scratch programming, improving engagement and problem-solving. Future research may examine how scaffolded tactics like peer mentorship or gamification can improve collaboration.

The Extent of Teamwork and Communication Contribute to Learners' Collaborative Experiences in Scratch Programming

Table 5 Extent of Teamwork and Communication in Collaborative Coding

Teamwork and Communication in Collaborative Coding	Weighted Mean	Verbal Interpretation
1. Collaborative Scratch projects have improved my ability to communicate and share ideas with my teammates.	3.71	To a great extent
2. I am more comfortable working with others to solve coding challenges in Scratch.	3.49	To a great extent
3. Teamwork during Scratch projects allows us to divide tasks and complete projects faster.	3.57	To a great extent
4. I have learned new coding techniques from my classmates during collaborative Scratch activities.	3.65	To a great extent
5. Collaborative coding has taught me how to listen to and incorporate others' ideas into our Scratch projects.	3.76	To a great extent
Overall Mean	3.64	To great extent

Table 5 displays the extent of teamwork and communication in collaborative coding among students in Scratch programming, using a weighted mean and its corresponding verbal interpretation. Teamwork and communication were crucial to Grade 7 students' Scratch programming collaborations, according to the poll. The weighted mean of 3.64, meaning "to a great extent," suggests that coding helps learners collaborate. Survey Question 5 (weighted mean = 3.76) showed that students strongly agreed that coding together helped them listen to and use their classmates' views. Interaction improves learning (Vygotsky, 1978). Survey Question 1 (weighted mean = 3.71) also demonstrated improved communication. Brennan & Resnick (2012) found that Scratch programming teaches pupils computational thinking and teamwork.

Survey Question 4 (weighted mean = 3.65) suggests that peer learning helped students learn new coding abilities. This supports the idea that learning together promotes knowledge sharing (Johnson & Johnson, 2014). Survey Question 3 (weighted mean = 3.57) demonstrated that delegating jobs helps projects operate more smoothly, demonstrating the importance of coordinated problem-solving.

Scratch-based collaborative coding increased seventh graders' collaboration and communication. All survey questions had high weighted averages, indicating that students learned best when they solved issues together, learned from each other, and split tasks. This supports previous research indicating collaborative programming improves technical and interpersonal skills (Grover & Pea, 2013).

Triangulation Of Data Analysis

The extent of teamwork and communication contributes to the Grade 7 learners collaborative experiences with Scratch programming, to their performance in the 3rd quarter assessment, and to their responses to open-ended survey questions.

The study found that teamwork and communication improved Grade 7 students' Scratch programming experiences. Statistics from exams, surveys, and performance reviews support this.

Students significantly agreed that coding collaboration improved peer listening, idea sharing, and task delegation (weighted mean = 3.64) in the survey. Zhang et al. (2020) found that structured peer interactions in programming improve problem-solving through collaborative debugging and information sharing. Grover et al. (2021) noted that real-time feedback and role-based job allocation improve communication on digital collaborative platforms like Scratch on Chromebooks.

Qualitative replies confirmed these findings. Students said collaboration helped them learn new coding approaches from peers and solve problems faster through brainstorming. These findings confirm Brennan and Resnick's (2021) claim that programming collaboration improves logic and social thinking. The third quarterly assessment linked effective cooperation to good and consistent problem-solving (MPS = 89.30%, SD = 0.2787). Well-structured group coding tasks diminish performance discrepancies, as shown by Kalelioglu and Gülbahar's (2023) minimal score variance.

However, numerous students found group dynamics difficult, pointing out the value of planned peer mentorship like "Debugging Captains" and gamified collaboration like "Bug Hunt Fridays" (Kafai & Burke, 2024). Bers et al.'s (2023) unplugged computational thinking tasks, such as flowchart preparation before coding, may improve mixed-ability group communication.

The triangulated data analysis shows that collaborating and communicating made Scratch programming more fun and effective, improving engagement and problem-solving. Future implementations should assign duties to group members and engage in multidisciplinary initiatives to optimize these benefits.

This connection shows that working together in an organized way turns coding into a learning experience that involves social interaction, which fits with socio-constructivist ideas (Vygotsky, 1978) and helps tackle current teaching challenges.

The Effectiveness of Collaborative Learning with Chromebooks in Enhancing Learners' Problem-Solving, Creativity, and Critical Thinking in Scratch Programming

Table 6 Effectiveness of Collaborative Coding with Chromebooks

Overall Effectiveness of Collaborative Coding with Chromebooks	Weighted Mean	Verbal Interpretation
1. Collaborative coding in Scratch, supported by Chromebooks, makes learning programming more engaging.	3.59	Very High Effectiveness
2. Working in a group helps me explore new ways of solving problems that I wouldn't have thought of alone.	3.59	Very High Effectiveness
3. Using Chromebooks during collaborative Scratch activities has made the coding process more enjoyable and efficient.	3.56	Very High Effectiveness
4. Collaborative coding encourages critical thinking and helps me find creative solutions to problems in Scratch programming.	3.57	Very High Effectiveness
5. I believe that collaborative Scratch projects with Chromebooks are an effective way to develop problem-solving skills.	3.61	Very High Effectiveness
Overall Mean	3.58	Very High Effectiveness

Table 6 displays the effectiveness of collaborative coding with Chromebooks among students in Scratch programming, using a weighted mean and its corresponding verbal interpretation. Chromebooks for group learning improved Grade 7 learners' use of Scratch programming for problem-solving, creativity, and critical thinking, according to studies. The weighted mean of 3.58, meaning "very high effectiveness," indicated favorable outcomes. Survey Question 5 (weighted mean = 3.61) demonstrated that students strongly believed Chromebook Scratch projects strengthened their problem-solving skills. Wing (2006) found that structured, interactive learning develops computational thinking.

Survey Question 2 (weighted mean = 3.59) showed that working in groups gave students fresh problem-solving ideas. This supports the idea that learning from others improves thinking (Vygotsky, 1978). Question 4

(weighted mean = 3.57) showed that students think creatively and critically when they work together. Bers et al. (2019) found that group coding improves logic and creativity.

Student responses to Survey Question 3 (weighted mean = 3.56) showed that Chromebooks made coding more fun and productive. This supports the 1:1 device program study (Penuel, 2006) that found making technology easier to use increases student engagement and productivity in digital learning environments.

Seventh graders improved their problem-solving, creativity, and critical thinking by coding using Chromebooks. High weighted averages for all survey categories showed that children learned best in a collaborative, technological environment. These findings support constructionist learning theories (Papert, 1980) that experiential, collaborative learning improves computational and advanced cognitive skills.

Triangulation Of Data Analysis

The effectiveness of collaborative learning with Chromebooks in enhancing learners' problem-solving, creativity, and critical thinking in Scratch programming, to their performance in the 3rd quarter assessment, and to their responses to open-ended survey questions.

In Scratch programming, collaborative coding on Chromebooks improved Grade 7 students' problem-solving, creativity, and critical thinking. Quantitative, qualitative, and performance data demonstrated this.

The survey (weighted mean = 3.58) found that most students agreed that Chromebook cooperation improved computational thinking and creative problem-solving. Zhang et al. (2020) found that real-time peer feedback improves logical thinking and creativity in digital coding environments. Cloud-based collaboration tools like Chromebooks enable iterative design thinking and problem-solving, according to Grover et al. (2021).

The third quarterly assessment (MPS = 89.30%, SD = 0.2787) showed that the intervention was beneficial, as pupils' problem-solving scores were high. The low standard deviation suggests that students learn similarly, supporting Kalelioglu and Gülbahar's (2023) claim that controlled group coding reduces achievement gaps.

Qualitative responses confirmed these findings. Students claimed teamwork "sparked new ideas" and "made debugging easier by combining different points of view." Brennan and Resnick (2021) found that collaborating on programming projects develops technical and artistic skills.

However, other students said uncontrolled teamwork caused off-task behavior. More advanced methods like role-specific peer mentoring ("Debugging Captains") and gamified challenges are needed (Kafai & Burke, 2024). Students can develop autonomous critical thinking before collaborative coding by using Bers et al.'s (2023) unplugged exercises like flowcharts.

Chromebooks for collaborative learning improved problem-solving, creativity, and critical thinking. Technology and peer connection facilitated aid. Future implementations should combine social learning and structured individual assignments for best performance.

This triangulation shows that using Chromebooks for collaborative work coincides with constructivist education (Papert, 1980), making Scratch programming a compelling 21st-century talent.

The Performance of the Students in Scratch Programming Based on their 3rd Quarter Assessment

Table 7 3rd Quarter Performance of the Learners in Scratch Programming

Grade & Section	N	HPS	LSO	HSO	MEAN	SD	MPS	No. of Cases Scored 75% and above	CLASS MASTERY
7-Archimedes	30	50	42	49	44.83	1.93	89.67%	30	100
7-Aristotle	33	50	41	49	44.70	2.17	89.39%	33	100
7-Copernicus	32	50	39	49	44.59	2.65	89.19%	32	100
7-Dalton	33	50	39	49	44.30	2.08	88.61%	33	100

7-Galileo	32	50	38	49	44.56	2.44	89.13%	32	100
7-Pascal	31	50	39	48	44.90	1.99	89.81%	31	100
	191	50	38	49	44.65	0.2787	89.30%	191	31.8333

Table 7 displays the third-quarter performance of the learners in Scratch programming, using a weighted mean and its corresponding verbal interpretation. The 50-item third-quarter assessment of 191 Grade 7 students' Scratch programming problem-solving skills is insightful.

The Mean Score (44.65 out of 50): Students demonstrated strong Scratch problem-solving skills with an average score of 44.65. The Mean Percentage Score (MPS of 89.30%) indicates that most students used computational thinking and collaborative coding methodologies. Score Range (Lowest: 38; Highest: 49): The narrow range suggests that most students performed well, with few outliers. Low standard deviation (0.2787): Scores grouped around the mean, indicating learner performance stability.

The low standard deviation (0.2787) and high mean percentage score (89.30%) show that collaborative Scratch coding on Chromebooks improved students' problem-solving skills. Collective coding improves computational thinking and problem-solving through peer learning and instantaneous feedback (Zhang et al., 2020). Chromebooks enabled real-time collaboration and accessibility, supporting Grover et al.'s (2021) findings that digital technologies boost programming teaching engagement and learning.

The low variance in outcomes (SD = 0.2787) suggests the teaching strategy is successful across the class, reducing performance discrepancies. Kalelioglu and Gülbahar (2023) found that planned, cooperative programming assignments improve learning fairness.

Data reveals that Grade 7 students solved issues better while using Chromebooks for collaborative Scratch coding. The strategy helped students think mathematically and help each other learn, as shown by the high mean score (44.65) and mean percentage (89.30%). The low standard deviation supports the hypothesis that the intervention reduced performance gaps for all students. These findings corroborate constructionist learning theories (Papert, 1980) that experience and collaborative learning improve computational and higher-order cognitive skills.

The Relationship Between the Effectiveness of Collaborative Learning with Chromebooks in Enhancing Learners' Problem-Solving Creativity and Critical Thinking and their 3rd Quarter Assessment in Scratch Programming

Table 8 Relationship between the effectiveness of collaborative learning with Chromebooks and 3rd quarter assessment in scratch programming

Effectiveness of Collaborative Learning	Pearson's r	t-value	p-value	Decision	Interpretation
3 rd Quarter Assessment	0.23	2.13	0.035	Reject Ho	Significant

The table shows how using Chromebooks for collaborative learning affects students' creativity and critical thinking in problem-solving and their scores in Scratch programming during the 3rd quarter, using the Pearson Product Moment Coefficient of Correlation (Pearson's r). The computed r-value of 0.23 indicates a weak degree of relationship. It suggests a slight tendency for the above variables to move in the same direction, but the relationship is not strong. However, the t-value used to test the significance of r showed a result of 2.13 (p=.035). Since the p-value is less than the .05 level of significance, the null hypothesis is rejected, which means that the relationship between the variables is statistically significant even if the strength of the relationship is weak. This finding is attributed to the large sample size. It can be concluded that as the perception of the learners on the effectiveness of Chromebooks in enhancing their problem-solving creativity and critical thinking, their 3rd quarter assessment scores in Scratch programming also get higher.

Pearson correlation analysis (*r* = 0.23, p = 0.035) demonstrates a weak but statistically significant association between utilizing Chromebooks to code together and improving Scratch programming evaluation ratings. Although the impact size is small, the significance (*p* < .05) indicates a non-random trend. This

supports earlier findings that digital collaborative learning environments like Scratch on Chromebooks can improve computational thinking and problem-solving, but the effect depends on lesson design and student interest (Grover et al., 2021; Zhang et al., 2020).

The weak connection ($r^* = 0.23$) may be due to:

1. Different groups may have collaborated better (Denner et al., 2021).
2. Limitations: The 50-item exam may not adequately assess creativity and critical thinking improvement.
3. Learners' motivation and coding experience may alter results (Kalelioglu & Gülbahar, 2023).

Suggested Additional Strategies or Activities can be Implemented to Further Enhance Grade 7 Learners' Problem-Solving Skills in Scratch Programming Beyond the Use of Chromebooks and Collaborative Learning

Survey Question 1 Short Answer: How has working collaboratively with your classmates in Scratch programming helped improve your problem-solving skills?

Working collaboratively on Scratch programming improves Grade 7 learners' problem-solving skills, according to the survey. Many students claimed working in groups helped them perceive things from different perspectives and tackle coding difficulties more creatively (Resnick et al., 2020). One student noted that a classmate's suggestion to use sine wave physics for jumping mechanics inspired a new computational approach. Brennan and Resnick (2021) found that working on Scratch projects together helps students create shared debugging processes to detect and repair flaws, improving their logical reasoning.

Students also said peer feedback helped them learn keyboard shortcuts, enhance scripts, and improve game mechanics (Grover & Pea, 2023). Brainstorming to solve difficulties was a common topic. A group used broadcast messages and variable checks to resolve a score-tracking program error. According to Vygotsky's (2020) social constructivist theory, peer interaction deepens learning. However, other students worked alone because they considered group dynamics to be inefficient (e.g., diverse ideas or unequal participation). This hypothesis suggests formal collaborative arrangements (Kafai & Burke, 2024).

Survey Question 2 Short Answer: In what ways do you think Scratch programming has helped you develop problem-solving skills that you can use in other subjects or real-life situations?

Scratch programming has considerably enhanced seventh graders' problem-solving skills, according to the survey. Many learners claimed they could apply their knowledge elsewhere and in life. Math and science benefit by breaking down tough problems into smaller, more doable steps (Grover & Pea, 2023). One student compared Scratch debugging to methodically uncovering flaws in arithmetic problems, which improves logic (Brennan & Resnick, 2021). Student discussions included tenacity and iterative thinking. A pupil commented, "Just like in life, Scratch teaches you to take on problems on your own and be patient." This finding matches computational persistence research (Kafai & Burke, 2024).

Students learned creative problem-solving by employing Scratch if-then-else logic to create real-world decisions (Resnick et al., 2020). Another student said debugging techniques like isolating problems and testing ideas helped them address non-coding technical difficulties. Skills can be applied elsewhere (Hsu et al., 2022). A small number of students felt Scratch only worked in ICT environments, suggesting metacognitive comprehension of skill use differs (Bers et al., 2023).

Survey Question 3 Short Answer: In what ways has the use of Chromebooks enhanced your learning and teamwork in Scratch programming?

Chromebooks make Scratch programming easier to access, interact with in real time, and work efficiently, according to the poll. Many students said Chromebooks' cloud-based nature makes sharing projects and getting feedback easy. These promote collaborative problem-solving (Grover & Pea, 2023). One student stated Chromebooks let their team "work on shared Scratch projects using cloud storage, instantly test each other's code, and use online resources to solve coding challenges together." This supports the premise that collaboration through technology improves logic (Brennan & Resnick, 2021).

Students reported Chromebooks made debugging and iteration easier because they are portable and can rapidly access online courses (Kafai & Burke, 2024). One stated, "The Chromebook's quick setup and easy-to-use interface help keep the focus on coding and creativity." According to studies, 1:1 device programs simplify digital learning (Hsu et al., 2022). But some students experienced issues with their browsers not working or their preference for other devices. The evidence suggests that infrastructure and device knowledge are crucial for implementation (Bers et al., 2023).

Overall General Analysis

The evidence shows that working together to code in Scratch on Chromebooks greatly increases problem-solving skills through peer learning, debugging, and creative brainstorming (Brennan & Resnick, 2021). But we need to use more scaffolding methods to make these skills even better. Structured peer mentoring, where more experienced students help less experienced students address problems, may help them become better at solving problems (Grover & Pea, 2023). Furthermore, adding unplugged computational thinking tasks, such as making flowcharts before coding, can help people develop better at breaking things down logically (Bers et al., 2023). Timed debugging races, or "bug hunt" competitions, are examples of gamified challenges that could make people more interested while also helping them learn how to troubleshoot in a systematic way (Kafai & Burke, 2024).

Differentiated pathways should be available for students who choose to work alone, such as optional paired debugging sessions or reflective journaling on how they solve problems (Hsu et al., 2022). Furthermore, Scratch projects that bridge disciplines, like making animations of science concepts or modeling math problems, could help students learn problem-solving skills that they can use in other areas (Resnick et al., 2020).

Chromebooks and working together are beneficial, but adding peer mentorship, unplugged activities, gamification, and projects that cross subjects will make the problem-solving programs even better. Based on the students' answers, here are some tactics or activities that can be used in addition to Chromebooks and group learning to help Grade 7 students improve their Scratch programming problem-solving skills even further. (1) Utilize peer mentorship roles, like "Debugging Captains," to enhance group learning. (2) Before coding, do some unplugged computational thinking tasks to help with planning. (3) Create gamified challenges, like "Bug Hunt Fridays," to encourage people to solve problems over and over again. (4) Give each learner alternatives, such as solo reflection journals. (5) Create Scratch projects that utilize skills from various subjects to demonstrate their practical applications.

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

The study concluded that Chromebook collaborative Scratch coding increased Grade 7 learners' critical thinking, problem-solving, and teamwork. Similar strategies were suggested for ICT courses, and technology-driven collaborative learning improved student engagement and skill development. This section shows a detailed discussion of the conclusions and recommendations drawn from the study.

Summary

Working together in Scratch boosted Grade 7 problem-solving in this study. Students learn to troubleshoot, reason, and be confident when facing difficult problems. Intervention improved problem-solving, pattern-finding, and task breakdown. The high survey ratings showed that peer cooperation boosted teamwork, communication, and problem-solving. Chromebook allows coding, teamwork, and real-time feedback. This method corrected ability gaps in learners with 89.30% computational thinking success and low standard variance.

Structured group coding and guided reflection help students solve difficulties. Encourage Scratch-based professional development. Learn challenging programming basics through collaborative coding. Pair programming, multi-step debugging, and creative Scratch projects help students solve challenges. Teachers should master Chromebook project-based learning and group coding. Peer mentoring and scaffolding help

underprivileged students. Timed debugging races engage pupils. Transdisciplinary Scratch projects like animating science issues should teach students how coding is utilized in real life, and long-term research should examine how well people retain their abilities.

Programming in Scratch on Chromebooks taught students to collaborate, solve issues, and try new things. Gamification, peer mentoring, and unplugged computational thinking exercises like flowchart creation may improve skills. Many students benefit from reflection journals and other methods. Students can learn professional problem-solving from many Scratch projects. Add these tactics to improve the classroom environment, engagement, and computational thinking in grade 7.

Conclusions

1. Grade 7 problem-solving skills increased after Scratch. Collaborative programming enhanced students' debugging skills, rational reasoning abilities, and confidence when facing challenging problems.
2. Grade 7 students have advanced Scratch programming problem-solving skills in deconstruction, debugging, and pattern recognition. Collaboration on Chromebooks boosted their confidence and readiness to try new solutions. These findings imply Scratch and peer learning can teach early secondary students computational problem-solving.
3. Chromebooks and Scratch collaboration improved Grade 7 students' creativity and critical thinking. The intervention improved problem-solving flexibility, logical thinking, and imaginative experimentation with high results across all domains.
4. Chromebooks helped Grade 7 students program Scratch, a study finds. Students stated Chromebooks helped with coding, teamwork, and communication. Chromebooks encouraged collaborative coding, as indicated by high mean scores across all survey items.
5. Results indicated that Scratch collaborative coding increased Grade 7 students' teamwork and communication. All survey questions received high weighted averages, indicating that cooperative problem-solving, peer learning, and task distribution benefited students.
6. Collaborative coding with Chromebooks improved Grade 7 students' problem-solving, creativity, and critical thinking. All survey questions had high weighted averages, suggesting that peer collaboration and technological integration generated an excellent learning environment.
7. Chromebook-based Scratch collaborative coding enhances Grade 7 students' problem-solving, research reveals. This method improved computational thinking and peer-assisted learning with a high mean (44.65) and percentage (89.30%). The intervention reduced the performance gaps among trainees, as indicated by a low standard deviation (SD).
8. Grade 7 Scratch skills and Chromebooks were moderately but significantly connected ($r^* = 0.23$, $p = 0.035$). Group Chromebooks improved student achievement. Collaboration in digital settings promotes computational thinking, but course design and student interaction are key. Group work, test limitations, and student coding may have reduced the link.
9. The implementation of collaborative coding in Scratch using Chromebooks resulted in the learners' improved problem-solving skills. These methods helped students solve problems rationally and fit varied learning styles, improving computational thinking in groups and individuals.

Recommendations

1. Teachers should integrate more structured collaborative coding tasks with guided reflection sessions and obtain Scratch-based collaborative learning professional development to engage students. Future studies could examine how collaborative coding develops complex programming ideas.
2. Scratch projects that need multi-step debugging and inventiveness improve problem-solving. Peer learning efficacy studies demonstrate pair programming may foster collaborative problem-solving. Scratch programming may encourage algorithmic thinking across grades in future research.
3. Regularly teaching Scratch-based collaborative coding boosts creativity and critical thinking. Collaborative coding session facilitation training may boost student engagement.

4. Chromebooks should promote programming and project-based class teamwork. Training teachers on how to use Chromebooks collaboratively may improve student learning. Understanding long-term effects on problem-solving and collaboration platforms can help establish the best programming training tools.
5. Teamwork can be improved by assigning coder, debugger, and presenter tasks. Reflective discussions help students recount collaborative adventures after exercises. Build task difficulty gradually to keep teamwork and engagement.
6. Guided peer review improves critical analysis of coding projects. Create Scratch projects that involve imaginative problem-solving. Blended learning approaches use collaborative coding and reflection exercises to promote autonomous thinking.
7. Differentiate instruction for lower performers by using scaffolding and peer mentoring, which help close modest performance gaps, achieving a success rate of 38 out of 50 trainees (76%). Multi-quarter longitudinal studies: Evaluate programming retention and skill transfer.
8. Group roles, peer review, project-based assessments, and cooperative coding should encourage creativity and critical thinking for problem-solving. Collaborative coding's long-term effects on problem-solving growth need a more comprehensive examination and longitudinal methodology.
9. The following methods may help Scratch programmers solve problems more effectively than using Chromebooks or collaborating with others: 1. Flowcharts and pseudocode encourage thinking before coding; 2. "Debugging Captains" refers to a collaborative peer mentoring approach; 3. Timed "Bug Hunt" tournaments promote systematic problem-solving; 4. Private reflection journals facilitate personal reflection.

Reflection

As a teacher-researcher, I learned that having my Grade 7 students work together on Scratch coding projects on Chromebooks helped them get better at solving problems. The intervention improved problem-solving, computational thinking, working together, and confidence in challenging tasks. Student interactions, fixing mistakes, and finding new ways to solve problems all demonstrated that collaborative learning in programming works. The high success rate (89.30%) and low performance variance revealed that structured group practice and guided reflection helped learners of all skill levels close gaps.

The students encountered problems with time management and group dynamics; people had to be given roles like coder and debugger, and scaffolding was needed for those who were having trouble. The positive survey responses revealed how Chromebooks made it possible for people to work together in real time, although other students needed a push to become involved. I now see how important it is to keep people interested by mixing collaborative coding with gamification and projects that cross disciplines. This study made it clear that professional development in collaborative tactics and technology integration is important. I hope that future research looks at how well people keep their skills over time. The program highlighted how working together on Scratch can make tackling problems fun and provide young learners more power.

REFERENCES

1. Anderson, L. W., & Krathwohl, D. R. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Longman.
2. Braun, V., & Clarke, V. (2021). Reflecting on reflexive thematic analysis. *Qualitative Research in Psychology*, 18(3), 328-344. <https://doi.org/10.1080/14780887.2020.1768816>
3. Dela Cruz, R. (2022). The Impact of Chromebook Use on Student Collaboration and Learning Outcomes in Middle School Classrooms. *Philippine Journal of Educational Research*.
4. Field, A. (2022). *Discovering statistics using IBM SPSS Statistics* (6th ed.). SAGE Publications.
5. Grover, S., & Pea, R. (2021). Computational thinking: A competency whose time has come. *Educational Researcher*, 50(7), 472-480. <https://doi.org/10.3102/0013189X211041528>
6. Creswell, J. W. (2021). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.

7. Hwang, G. J., Lai, K. W., & Wang, S. Y. (2021). Seamless flipped learning: A mobile technology-enhanced learning environment for developing students' collaborative problem-solving skills. *Computers & Education*, 164, 104114. <https://doi.org/10.1016/j.compedu.2020.104114>
8. Kafai, Y. B., & Burke, Q. (2020). *Connected Code: Why Children Need to Learn Programming*. MIT Press.
9. Maligalig, J., & Tan, C. (2020). Collaborative Learning and Problem Solving in ICT Education. *Journal of Educational Technology*.
10. Mertler, C. A. (2020). *Action research: Improving schools and empowering educators* (5th ed.). SAGE Publications.
11. Pierson, T., & Williams, B. (2022). Collaborative learning in the digital age: The role of technology in fostering 21st-century skills. *Journal of Digital Education*, 15(2), 89-104. <https://doi.org/10.1080/21532974.2022.1488963>
12. SEAMEO. (2021). *Regional Comparative and Explanatory Study on Student Achievement in STEM*. Southeast Asian Ministers of Education Organization.
13. Wastiau, P., & Rodríguez, D. (2021). Collaborative coding in the classroom: Leveraging technology to enhance student learning outcomes. *Computers & Education*, 167(1), 104-112. <https://doi.org/10.1016/j.compedu.2021.104112>
14. Zhong, L., & Song, Y. (2023). Enhancing critical thinking and collaboration through coding: A case study of middle school students. *Journal of Computer Education*, 10(1), 43-58. <https://doi.org/10.1007/s40692-023-00268-x>
15. Atmatzidou, S., & Demetriadis, S. (2022). Enhancing students' computational thinking skills with educational robotics: A systematic review. *Computers & Education*, 178, 104382.
16. Brennan, K., & Resnick, M. (2020). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 14th International Conference on Learning Sciences*, 1-3.
17. Grover, S., & Pea, R. (2023). Computational thinking: A competency whose time has come. *Computer Science Education*, 33(1), 1-24.
18. Hsu, T. C., Chang, S. C., & Hung, Y. T. (2021). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126, 296-310.
19. Kong, S. C., & Wang, Y. Q. (2020). Collaborative programming in Scratch: A study on students' learning engagement. *Journal of Educational Computing Research*, 58(3), 570-595.
20. Zhang, L., Nouri, J., & Mannila, L. (2021). *A systematic review of learning computational thinking through Scratch in K-9.* *Computers & Education*, 173, 104270.
21. Brennan, K., & Resnick, M. (2020). *Scratch: A framework for assessing computational thinking development*. MIT Media Lab.
22. Denner, J., Werner, L., & O'Connor, L. (2021). Pair programming in middle school: What does it take to improve problem-solving skills? *Journal of Educational Computing Research*.
23. Grover, S., & Pea, R. (2023). *Computational thinking in K-12: A review of the literature*. *Computer Science Education*.
24. Hsu, T.-C., Chang, S.-C., & Hung, Y.-T. (2021). How to learn and how to teach computational thinking: Suggestions based on a review of literature. *Computers & Education*.
25. Kong, S.-C., Wang, Y.-Q., & Lai, M. (2022). Effects of computational thinking on problem-solving performance in programming. *Journal of Educational Computing Research*.
26. Lye, S. Y., & Koh, J. H. L. (2020). Review on teaching and learning of computational thinking through programming. *Computers in Human Behavior*.
27. Grover, S., & Pea, R. (2023). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 52(1), 38-54.
28. Kong, S. C., et al. (2022). Creativity in block-based programming: A systematic review. *Computers & Education*, 180, 104430.
29. Sáez-López, J. M., et al. (2020). Visual programming languages integrated across the curriculum in elementary school. *Computers & Education*, 158, 103989.
30. Zhang, L., & Nouri, J. (2021). A systematic review of learning computational thinking through Scratch. *IEEE Transactions on Education*, 64(3), 367-378.

31. García-Martínez, I., Fernández-Batanero, J. M., Cobos Sanchiz, D., & Luque de la Rosa, A. (2023). Using digital tools to foster collaborative learning in programming: A systematic review. *Computers & Education*, 184, 104521.
32. Hwang, G. J., & Fu, Q. K. (2020). Advancing collaborative learning with technology: A meta-analysis. *Educational Research Review*, 30, 100337.
33. Lee, J., & Kim, H. (2021). Enhancing programming education through collaborative coding tools. *Journal of Educational Computing Research*, 59(5), 789-812.
34. Smith, A., Jones, B., & Brown, C. (2022). *The impact of cloud-based devices on K-12 collaborative learning.* *TechTrends*, 66(3), 412-425.
35. Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking.
36. Johnson, D. W., & Johnson, R. T. (2014). *Cooperative learning in 21st-century classrooms.*
37. Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.*
38. Bers, M. U., et al. (2019). Coding as a playground: Computational thinking in early childhood.
39. Hattie, J., & Timperley, H. (2007). The power of feedback.
40. Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas.*
41. Resnick, M. (2017). *Lifelong Kindergarten: Cultivating creativity through projects, passion, peers, and play.*
42. Brennan, K., & Resnick, M. (2022). *Scratch: A tool for developing computational thinking.* MIT Media Lab.
43. Denner, J., Werner, L., & Ortiz, E. (2021). Pair programming in middle school: What does it look like? *Journal of Educational Computing Research*.
44. Grover, S., Pea, R., & Cooper, S. (2021). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*.
45. Kalelioglu, F., & Gülbahar, Y. (2023). The effects of teaching programming via Scratch on problem-solving skills. *Informatics in Education*.
46. Zhang, L., Nouri, J., & Mannila, L. (2020). A systematic review of learning computational thinking through Scratch. *IEEE Transactions on Education*.
47. Brennan, K., & Resnick, M. (2021). *ScratchEd: Computational Thinking and Creativity in Learning Communities.* MIT Press.
48. Grover, S., & Pea, R. (2023). "Collaborative Coding in K-12: A Meta-Analysis of Problem-Solving Outcomes." *Journal of CS Education*, 14(2).
49. Kafai, Y., & Burke, Q. (2024). *Connected Code: Why Children Need to Learn Programming.* MIT Press.
50. Vygotsky, L. (2020). *Mind in Society: Social Constructivism in Learning.* Harvard UP.
51. Bers, M., et al. (2023). *Beyond Coding: How Children Learn Human Values Through Programming.* MIT Press.
52. Brennan, K., & Resnick, M. (2021). *ScratchEd: Computational Thinking in Creative Learning.* MIT Press.
53. Grover, S., & Pea, R. (2023). "Collaborative Coding in K-12: A Meta-Analysis of Problem-Solving Outcomes." *Journal of CS Education*, 14(2).
54. Hsu, T., et al. (2022). "1:1 Device Programs and Equity in Computational Participation." *TechTrends*, 66(3).
55. Kafai, Y., & Burke, Q. (2024). *Connected Code: Why Children Need to Learn Programming.* MIT Press.