applied sciences

MDPI

*Article*

# Developing Computational Thinking: Design-Based Learning and Interdisciplinary Activity Design

**Dongqing Wang [1,†], Liqiang Luo [2,†], Jing Luo [1,†], Sihong Lin [3] and Guangjie Ren [1,*]**

[1] School of Information Technology in Education, South China Normal University, 55 Zhongshan Dadao West, Tianhe District, Guangzhou 510631, China
[2] Primary School Affiliated to South China Normal University, 55 Zhongshan Dadao West, Tianhe District, Guangzhou 510631, China
[3] Dongguan Science Technology School, No.2, Zhijiao Road, Hengli Town, Dongguan 523000, China
* Correspondence: renguangjie@m.scnu.edu.cn
† These authors contributed equally to this work.

**Abstract:** As research progresses, integrating computational thinking (CT) and designing interdisciplinary activities to teach various disciplines have gradually emerged as new ideas and important ways to develop the CT of students. This paper introduces the concept of design-based learning (DBL) and analyzes the internal connections between DBL and CT teaching. In this study, an interdisciplinary activity design model was constructed based on an analysis of existing design-based scientific cycle models and research into STEAM education, which is an approach to learning that uses science, technology, engineering, the arts, and mathematics as access points for guiding student inquiry, dialogue, and critical thinking. Next, specific activities with a focus on CT were designed to teach graphical programming to fifth grade students using Scratch. This quasi-experimental research was carried out to test the promotion effects of interdisciplinary activity design and traditional programming activities on the CT of students. Finally, the results showed that the proposed interdisciplinary activity design could develop the CT levels of students more effectively than traditional programming activities.

**Keywords:** computational thinking; design-based learning; interdisciplinary activity design; graphical programs

check for updates

**Citation:** Wang, D.; Luo, L.; Luo, J.; Lin, S.; Ren, G. Developing Computational Thinking: Design-Based Learning and Interdisciplinary Activity Design. *Appl. Sci.* **2022**, *12*, 11033. https://doi.org/10.3390/app122111033

Academic Editor: Antonio Visioli

Received: 20 September 2022
Accepted: 25 October 2022
Published: 31 October 2022

## 1. Introduction

With the development of artificial intelligence, big data, and other emerging technologies, exploring practical paths for the development of the high-order thinking abilities of students has become a research hot spot in the study of school education. Computational thinking is regarded as a necessary skill for the intelligent era, which has been studied and defined by many researchers. CT was first introduced by Papert in 1996. Wing outlined the basic definition of CT as a way of "solving problems, designing systems and understanding human behavior by drawing on the concepts of computer science" [1]. Since then, CT has been considered by a growing number of researchers, especially within elementary education. The International Society for Technology in Education (ISTE) and the Computer Science Teacher Association (CSTA) defined the vocabulary for CT as: data collection, data analysis, data representation, problem decomposition, abstraction, and four other words [2]. CT has become a key digital literacy skill that digital citizens must master in the 21st century. Its popularization in K–12 education represents the inevitable trend of globalization [3].

As an element of the core information technology literacy discipline, CT has been included in the curriculum standards of senior high schools in China since 2017. Through the teaching reform of information technology courses in primary and secondary schools, researchers from China have gradually explored effective ways to develop CT in practice. With reference to the example of CT training in British school computing curricula, Xie Zhong-Xin and Cao Yang-Lu analyzed how to realize CT training in Chinese school information technology curricula [4]. Combined with the use of App Inventor, Guo Shou-Chao et al. proposed learning models for the processes of problem posing, abstract description, reduction, decomposition, modular methods, and module splicing programming to help students to solve problems in class [5]. Liang Yun-zhen et al. took Brennan's framework of CT [6] as a theoretical reference for sixth grade information technology classes in primary school, and proposed a human–computer collaborative precision teaching mode that was oriented toward the cultivation of CT. They used a computational thinking test, Dr. Scratch, evaluation scales, and other tools to test the levels CT among students [7].

With the in-depth study of STEAM education in recent years, how to organically embed CT into interdisciplinary teaching practice has gradually emerged as a new idea and an important way to develop CT. For example, in language learning, Burke used Scratch to develop the writing skills of students [8]. Zhou Ping-hong et al. developed a STEM engineering design teaching model that integrated CT. The application of this model to teaching the STEM course "Plant Factory" showed that the integration and practice of CT in each step of STEM engineering design could significantly improve the attitudes of students toward STEM and their CT ability [9].

In CT education, learners can create games to help them to learn more effectively. Specifically, since CT education is based on algorithms and programming education, learners acquire skills and knowledge through the process of making programs or games themselves [10]. SooJin Jun et al. considered how CT education could create learning environments by actively using DBL. They performed an experiment and the results indicated that DBL improved CT in their sample of elementary school students [11].

Therefore, on the basis of our systematic analysis of the internal connections between DBL and CT, we constructed an interdisciplinary activity design to develop the CT of students. We then verified the effectiveness of our design through quasi-experimental research. The research questions of this study were as follows:

1. How can interdisciplinary activities be designed to develop the CT of students?
2. How do interdisciplinary activities promote the CT of students?

## 2. Background

Different from traditional linear and centralized learning, interdisciplinary STEAM learning emphasizes the integration and application of knowledge from different subjects. STEAM focuses on problem-oriented cooperation and communication, as well as forming solutions to develop innovative talents [12]. The development of CT has offered the dual attributes of computational science and education, which not only reflects the characteristics of interdisciplinary learning, but also highlights the development modes of innovative talents.

### *2.1. The Evolution of the Concept of CT*

Since Papert first put forward the concept of "procedural thinking" in 1996, CT has developed over more than 20 years. During this period, many scholars and research institutions have defined this concept. In addition to the definitions from Wing, the ISTE, and the CSTA, Brennan and Resnick also proposed three key dimensions of CT, which are widely accepted. In summary, Brennan, Resnick, and other researchers expounded CT using an operational definition: the framework of CT includes concepts (sequences, loops, parallelism, events, conditionals, operators, and data), practices (being incremental and iterative, testing and debugging, and reusing and remixing, as well as abstracting and modularizing), and perspectives (expressing, connecting, and questioning) [13]. However, Fraillon, Ainley, and Schulz expounded CT using a more general definition: CT refers to an individual's ability to recognize aspects of real-world problems that are appropriate for computational formulation, and to evaluate and develop algorithmic solutions to those problems so that the solutions can be operationalized using a computer. This definition involves concepts such as conceptualization, data collection, data representation, planning evaluation, and implementation [14].

CT originated from the discipline of computational science, and its definition has mainly evolved from programming education. With the rise of STEAM education, interdisciplinary concepts have been introduced to the definition of CT. CT parallels the core practices of science, technology, engineering, and mathematics (STEM) education, and is believed to effectively support the learning of scientific and mathematical concepts [15]. In a system of Computational Thinking in STEM, CT includes modeling and simulation practices, computational problem-solving practices, and data practices and systems thinking practices, which are used to evaluate students' mastery of CT within the STEM discipline [16]. Until now, the concept of CT has been integrated into various disciplines.

### *2.2. The Strategy and Mode of Developing CT*

How to effectively develop CT has become the focus of current research. The authors of [17] conducted a meta-analysis of CT research in SSCI/SCI (Social Sciences Citation Index/Science Citation Index) journals from 2006 to 2017 and identified 13 kinds of CT training strategies. These strategies included problem-based learning, collaborative learning, project-based learning, game-based learning, scaffolding, storytelling, systematic computational learning, aesthetic experiences, concept-based learning, human–computer interaction teaching, embodied learning, universal design for learning, and design-based learning.

Researchers have carried out in-depth studies on these strategies. Farris [18] introduced music and ViMAP, combined with experiential teaching, into the teaching of fifth grade physics to train students' physical concepts and aesthetic experiences, and to train students to master CT methods and steps. Soleimania [19] used CyberPLAYce to organically integrate digital stories and physics, and verify its role in promoting CT. In addition, Mustafa [20] compared the effects of DBL activities and programming tasks on CT skills through similar experimental studies, and found that DBL activities were similar to programming tasks in terms of playing a positive role in promoting CT. In recent years, DBL has been considered as a critical approach to improving CT [21].

## 3. Methodology

### *3.1. DBL and CT Development*

DBL and CT development represents the relationship between means and objectives. They both emphasize the diversification of the teaching process: firstly, practical teaching content is helpful for the deep understanding of abstract concepts in CT as practical and interdisciplinary practices can help to make CT more concrete and visual; secondly, comprehensive interdisciplinary problems help students to deeply understand the process of problem-solving, since DBL focuses on "design" as the intermediary of inquiry learning, with the goal of changing existing situations into learning situations, thereby helping learners to carry out interdisciplinary problem-based learning and not only helping with the deeper understanding of abstract concepts, but also with experiences of problem-solving situations; thirdly, iterative learning processes are helpful for mastering CT methods; fourthly, product learning results are helpful for the visual expression of CT; finally, diversified teaching evaluations contribute to the comprehensiveness of CT evaluation, and the multiple teaching evaluations advocated by DBL provide a variety of ways to evaluate and improve CT.

### *3.2. Interdisciplinary Activities and CT*

Interdisciplinary activity design is a research trend within CT training. CT has broken through the boundaries of computational science and has started to be developed more extensively within other disciplines. Computational thinking should not be limited to computer programming, but should instead become one of the necessary skills for daily life, as with reading and writing [22,23]. CT can be transferred to fields other than programming (e.g., science, social sciences, humanities, etc.) [22,24], and can be combined with various disciplines [25], such as English [8,26], art [27], biology [28,29], and mathematics [30]. Snodgrass [31] introduced Scratch into primary school mathematics to promote complex mathematical problem-solving among disabled students and to develop their CT ability. Matsumoto and Cao applied Excel to high school chemistry teaching, and trained students' CT ability through simulations and modeling, experimental data analyses, and coding/programming and algorithm reasoning, as well as statistics and probability activities [32]. In addition, the development of the CT of students through the STEM teaching process of "determine the problem, plan, propose solutions, modify and communicate" has been proposed [33]. In terms of interdisciplinary curriculum design, different countries have explored new forms of CT curricula through interdisciplinary ideas. For example, the Finnish National Education Council turned programming into a whole discipline and formed a "computing" course to develop the CT of students [34], while Malaysia [35] incorporates CT, problem-solving, and information technology into all subjects in primary schools, and encourages students to use tools to solve complex practical problems so as to develop their CT and problem-solving abilities.

### *3.3. Design-Based Scientific Cycle Model and CT*

The design-based scientific cycle model was proposed by Kolodner, Paul, and David [36] according to the iterative cycle characteristics of DBL. In this model, there are two cyclic and iterative processes, namely design/redesign and investigate/explore. The process of design/redesign includes the understanding of the challenge, the planning process, and the design (using science). The process of investigate/explore mainly involves conducting investigations, including design investigations, and more. It should be noted that "need to do" and "need to know" are the links between the iterative processes of design/redesign and investigate/explore.

These iterative learning processes are the core features of the design-based scientific cycle model. Helping students to identify, analyze, and solve problems through the design/redesign and investigate/explore processes is the key element of the model. Different from traditional inquiry learning, DBL starts with projects and tasks and then promotes students' mastery of problem-solving processes through the iterative cycle of planning, designing, understanding the challenge, etc.

### *3.4. CT-Oriented Interdisciplinary Activity Design*

Combined with the training goal of CT and under the guidance of STEAM interdisciplinary teaching, we constructed an interdisciplinary activity design based on DBL [37], which took active learning as the goal and the cycle of discovery, design, and expression practices as the main teaching approach. In this model, the teacher's activities included proposing tasks and offering support and guidance, while the students' activities included clarifying, modifying, and optimizing those tasks. See Figure 1 for more detail.

Discovery practices are involved in the beginning of learning activities, and are key to determining benign problems. To facilitate discovery practices, teachers need to design and issue learning task sheets according to the teaching objectives. Different from traditional learning task sheets, these not only include the information technology subject knowledge required to solve the preset problem but also other subject knowledge related to the problem. Then, according to the task sheets, students complete previews before class. The teachers determine the students' original levels using these previews, answer questions in class to set up a gradient and real problem situation, and then help the students to solve the real problem. The purpose of this process is to help teachers to understand learning situations through self-study before class and to help students to develop the habit of autonomous learning and problem awareness.

The main purpose of design practices is to develop the problem-solving abilities of students. Firstly, teachers are required to guide students to decompose the problems that need to be solved and use multidisciplinary knowledge to design solutions. Secondly, teachers are required to organize groups to carry out research, employ creative learning, and share and exchange solutions; then, they need to guide students to optimize their solutions. If the solutions cannot solve the specific problems, then the students must return to the task decomposition stage, carry out the design practice again, and form new solutions through exploration and practice.

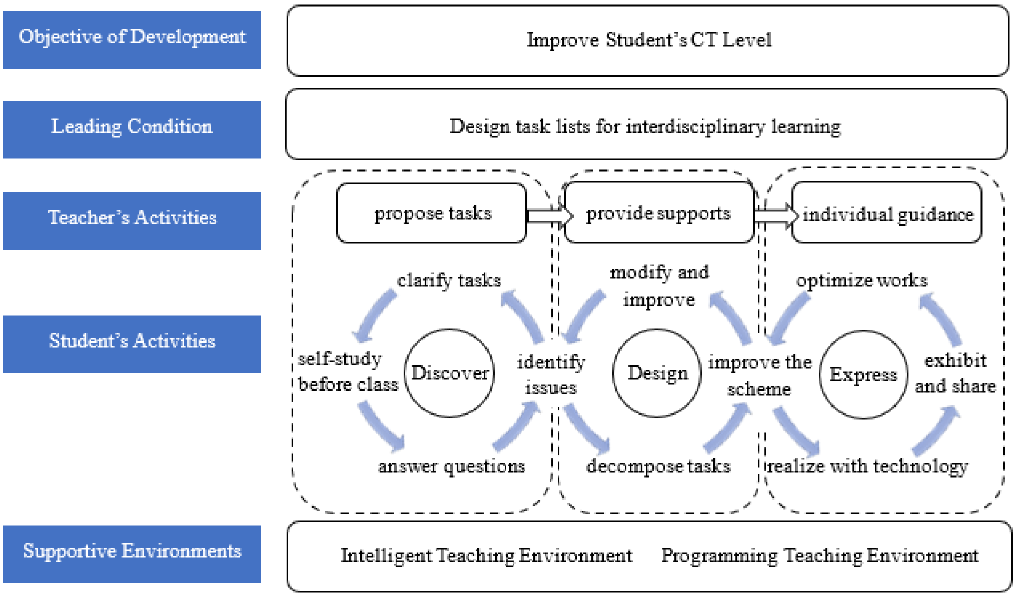


**Figure 1.** Our interdisciplinary activity design for the development of CT.

Expression practices require teachers to guide students to choose the appropriate technology implementation tools (e.g., Scratch, APP Inventor, etc.) for independent programming creations, and provide technical guidance to realize visual expressions of creativity. After developing independent programming creations, teachers must organize students to carry out group research and creative learning to allow them to share ideas with each other and put forward points to be improved. After that, the students should revise their creations according to the feedback. If students cannot complete their programming work according to solutions based on multidisciplinary knowledge, then they need to return to the design, exhibition, and sharing stages, and complete many design practices until they can complete their programming work. It should be noted that the knowledge of non-information technology disciplines that is required by students to solve these problems should not prevent them from carrying out scheme designs and work implementations. The main purpose of expression practices is to help students to combine information technology with other disciplines and develop their CT on the basis of enhancing a deep understanding of the application of subject knowledge by solving real and comprehensive problems using their subject knowledge and information technology literacy.

## 4. Implementation and Practice

### *4.1. Practice Plan*

Taking a Scratch course from an information technology textbook from Guangdong Education Press as an example, we were able to design and develop a teaching resource package for this study. The package served as an interdisciplinary activity to develop CT. A quasi-experimental study of 10 class hours was carried out to compare the effects of this interdisciplinary activity design and traditional programming activities on the CT of students. In the design of the teaching content, the experimental class adopted the interdisciplinary activity design that was oriented to the development of CT by reorganizing and integrating the teaching content of Scratch into five interdisciplinary themed activities: “I'm the story king”; “I'm the little test king”; “I'm the little operator”; “I'm the explorer”; and “I can get the average”. The same knowledge points were used for the control class. See

Table 1 for more detail. The teaching for the control class was based on Scratch curriculum content for traditional programming design and teaching. The CTt and Dr. Scratch were then selected as the evaluation tools for the practical effects.

**Table 1.** Teaching content of experimental class and control class.

| Class Type | Teaching Content | Required Disciplines | Cycle of Instruction |
|---|---|---|---|
| Experimental Class | 1. I'm Story King: use Scratch to complete Chinese composition writing<br>2. I am quiz king: use scratch to complete the English self test<br>3. I am an arithmetic expert: using Scratch to realize the design of addition, subtraction, multiplication, and division<br>4. I am an explorer: use Scratch to draw regular polygons<br>5. I can calculate the average: use Scratch to complete the program of average of a group of data | Information Technology<br>Chinese<br>English<br>Mathematics | 2 class hours, 10 class hours in total |
| Control Class | 1. Little magician: role and appearance<br>2. Happy Bouncing: Background, Rotation, and Moving<br>3. Funny shooting game: positioning and translation<br>4. Cool Cat Playing Football: Repetitive Execution and Control<br>5. Happy Pig Race: keyboard control and condition detection<br>6. IQ challenge: keyboard information acquisition and detection<br>7. I'll pick apples: role control and variables<br>8. Remote control helicopter: role orientation and stop execution<br>9. Interesting graphics: brush command<br>10. Making promotional films: broadcasting and receiving broadcasting | Information Technology | 1 class hour, 10 class hours in total |

Among them, the CTt (computational thinking test) is a set of CT test questions designed by Román-González et al., based on basic psychological and problem-solving abilities, which mainly involves the concepts of sequences, cycles, events, conditions, parallelism, operators, data, etc. [38]. It contains 28 items and takes 40 min to complete. Before the experiment, a CTt pre-test was carried out and the Cronbach's alpha reliability of the CTt was calculated as 0.815. The KMO validity (Kaiser–Meyer–Olkin) value was found to be 0.829, which indicated that the overall reliability and validity of the CTt was acceptable. The reliability and validity were good, so the test data were reliable.

### 4.2. Design and Practice of Teaching Activities

Our design highlighted "interdisciplinary teaching" and "DBL processes". It aimed to improve the programming and logical thinking of students as they solved complex problems, by guiding the students to use computer programming and multidisciplinary knowledge to develop their CT abilities. Combined with the interdisciplinary activity design for CT and according to the characteristics of the physical and mental development and learning levels of the students, specific activities on specific topics were designed to carry out the teaching activities for CT development. The following example is the specific activity for the teaching theme "I am an explorer: Discovering the mystery of regular polygons" (Figure 2).

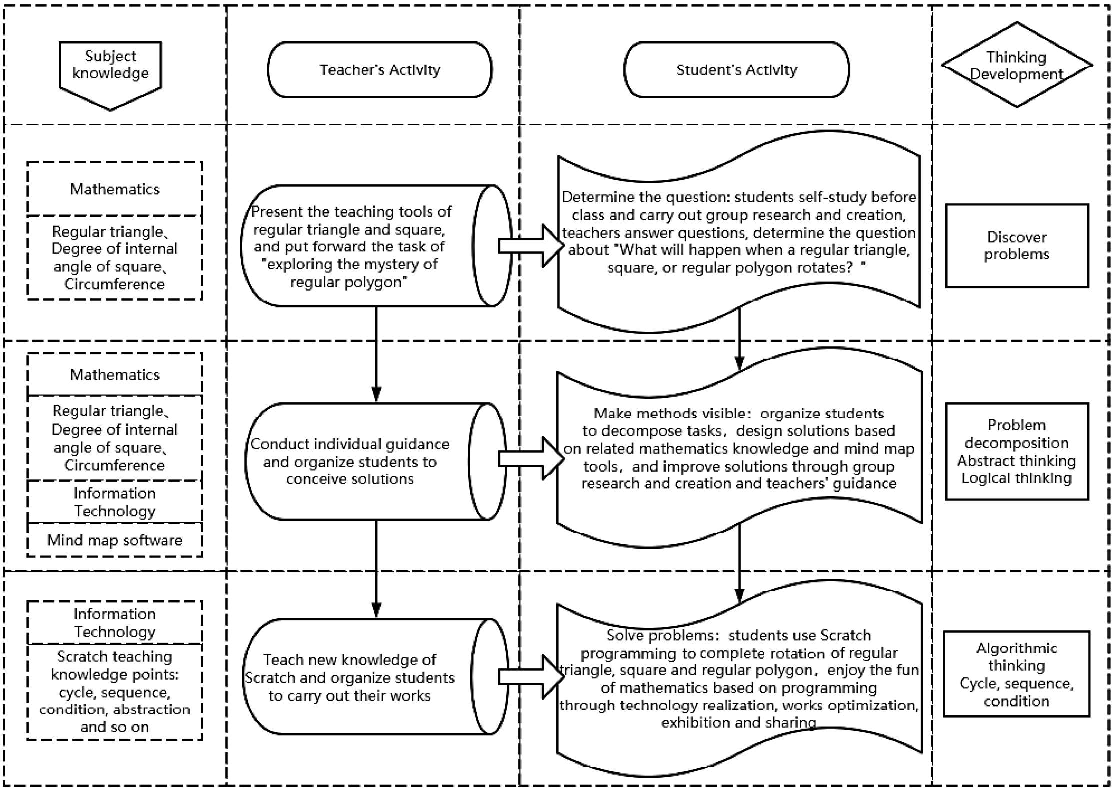
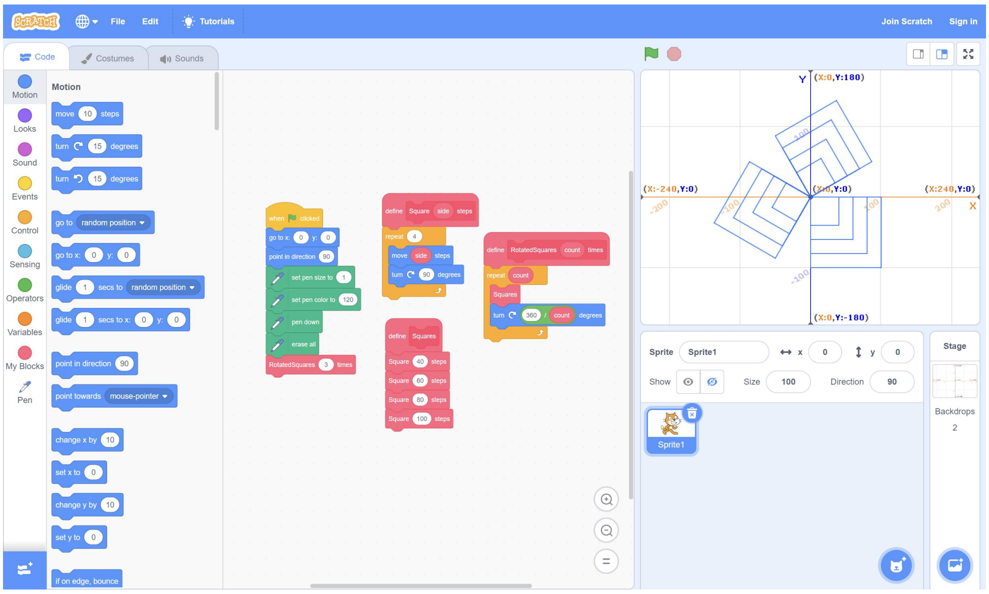

Subject knowledge
Teacher's Activity
Student's Activity
Thinking Development
Mathematics
Regular triangle、Degree of internal angle of square、Circumference
Present the teaching tools of regular triangle and square, and put forward the task of "exploring the mystery of regular polygon"
Determine the question: students self-study before class and carry out group research and creation, teachers answer questions, determine the question about "What will happen when a regular triangle, square, or regular polygon rotates? "
Discover problems
Mathematics
Regular triangle、Degree of internal angle of square、Circumference
Information Technology
Mind map software
Conduct individual guidance and organize students to conceive solutions
Make methods visible: organize students to decompose tasks, design solutions based on related mathematics knowledge and mind map tools, and improve solutions through group research and creation and teachers' guidance
Problem decomposition Abstract thinking Logical thinking
Information Technology
Scratch teaching knowledge points: cycle, sequence, condition, abstraction and so on
Teach new knowledge of Scratch and organize students to carry out their works
Solve problems: students use Scratch programming to complete rotation of regular triangle, square and regular polygon, enjoy the fun of mathematics based on programming through technology realization, works optimization, exhibition and sharing
Algorithmic thinking Cycle, sequence, condition

**Figure 2.** The specific activities for the teaching theme, "I'm an explorer: Discovering the mystery of regular polygons".

1. *Students determine the theme, investigate its background, explore independently, clarify the task, and initially form an awareness of the identified problems:* The core of our interdisciplinary activity design was to create real interdisciplinary application situations. Under the guidance of the teachers, the students independently found and proposed situations related to regular polygons. Through task decomposition, they used relevant mathematical knowledge, such as finding the degrees of internal angles and the perimeters of regular triangles and squares. The students also shared and communicated in group research and creation, and the teachers answered questions about the problems to be solved.
2. *Students draw up a plan, brainstorm ideas about the task, experience problem-solving processes, and develop their logical thinking:* After clarifying the task, the teachers needed to encourage the students to carry out self-exploration around how to solve the problems and use Scratch learning content to decompose problems, and design and improve solutions. Under the guidance of the teachers, the students learned certain knowledge points, such as cycles and sequence control, and used mind mapping tools to design solutions that promoted the visualization of their thinking process. After designing their solutions, the students were organized into groups to share their ideas and communicate with each other to optimize the solutions, thus stimulating the students' awareness of the importance of sharing and communication in developing abstract and logical thinking.
3. *Students creatively express, share, display, and perfect their works in combination with peer communication and teacher guidance, so as to promote their programming thinking and creativity:* After determining the optimal solution, the next task was to operate. The teachers helped the students to complete the tasks by providing technical guidance and solutions for students who were having difficulties. Next, the results were reported, with the students organized in groups to discuss the decomposition and form problem-solving ideas, which could not only provide guidance for students

with learning difficulties, but also help students to find the shortcomings and points to be improved in other works. Then, according to the feedback from the student groups, they revised and improved their works until the learning task was completed successfully. Finally, the teachers summarized the class content. After learning each topic, the teachers needed to summarize what the students had learnt, paying attention to student participation and the completion of tasks to help the students to understand their learning situations. Figure 3 shows the effects of some completed works, including modules built using Scratch and a schematic diagram of regular polygon rotation.



```
when 🏴 clicked
go to x: 0 y: 0
point in direction 90
set pen size to 1
set pen color to 120
pen down
erase all
RotatedSquares 3 times

define Square side steps
repeat 4
    move side steps
    turn ↻ 90 degrees

define Squares
Square 40 steps
Square 60 steps
Square 80 steps
Square 100 steps

define RotatedSquares count times
repeat count
    Squares
    turn ↻ 360 / count degrees
```

**Figure 3.** The effects of the work of students in the experimental class.

## *4.3. Research Results*

### 4.3.1. CT Performance

To explore the differences between the experimental group and the control group, the CT scores of the two groups were analyzed before and after the experiment, as shown in Table 2. It can be seen from the results that the CT levels of the two groups before the experiment were basically the same, and that there was no significant difference between the students' CT scores (Sig. = 0.899). After the experiment, there was a significant difference in CT score between the experimental group and the control group (Sig. = 0.018), with the CT score of the experimental group being significantly higher than that of the control group.

**Table 2.** Descriptive statistics and independent sample *t*-test of students' CT performances in experimental class and control class.

| | **Group** | **N** | **Mean** | **Standard Deviation** | ***t*** | **Sig (Bilateral)** |
|---|---|---|---|---|---|---|
| CT (pre-test) | Experimental Class<br>Control Class | 44<br>40 | 5.924<br>6.008 | 3.326<br>2.675 | −0.128 | 0.899 |
| CT (post-test) | Experimental Class<br>Control Class | 44<br>40 | 9.23<br>7.50 | 3.75<br>2.69 | −2.404 | 0.018 |

To further explore the changes in the CT scores of the students in the experimental and control groups before and after the experiment, the pre-test and post-test CT scores of the two groups were analyzed. The results are shown in Table 3. A paired sample *t*-test was carried out on the pre- and post-test data from the experimental group and it was found that the CT performance of the experimental group significantly improved after the experiment (Sig. = 0.000). At the same time, a paired sample *t*-test was carried out on the pre- and post-test data from the control group, and it was found that the CT performance of the control group also significantly improved after the experiment (Sig. = 0.027).

**Table 3.** Paired sample *t*-test of students' CT performances in experimental class and control class.

| | | N | Mean | Standard Deviation | df | *t* | Sig (Bilateral) |
|---|---|---|---|---|---|---|---|
| CT (Control Class) | pre-test<br>post-test | 40<br>40 | 5.92<br>7.50 | 3.33<br>2.69 | 39 | −2.294 | 0.027 |
| CT (Experimental Class) | pre-test<br>post-test | 44<br>44 | 6.01<br>9.23 | 2.68<br>3.75 | 40 | −4.938 | 0.000 |

A comparison of the changes in CT knowledge levels between the experimental class and the control class was performed using the post-test data (Figure 4). Scores closer to the edges indicated that the students had higher levels of mastery of CT, and scores that were closer to the middle indicate that the students had lower levels of mastery of CT. Through this analysis, we found that after the teaching experiment, the students in the experimental class had a better grasp of the orders, cycles, events, conditions, and other dimensions from the same level of pre-test CT.

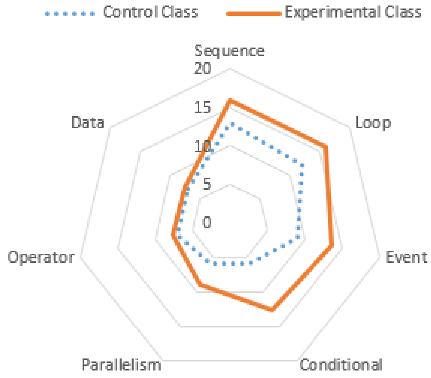


**Figure 4.** The results of post-test data on CT scores.

### 4.3.2. Practical CT Skills

Through the Dr. Scratch analysis of the students' completed works, we could measure the students' influence on their practical CT skills. The results showed that the parallelism (multiple scripts), loops (repeated execution), abstraction (defining new blocks), data representation (no operation on variables and tables). and synchronization (no simultaneous use of background and broadcast blocks) scores were very low. In the case of the same level of pre-test CT knowledge, after the teaching experiment was completed, the students in the experimental class had a better grasp of the abstract, problem-solving, logical thinking, synchronization, parallelism, and sequence control dimensions, among others. However, on the whole, these students achieved low scores in data representation, user interaction, and other dimensions. The reason for this might depend on the difficulty and emphasis of the interdisciplinary activity design. The themed design of the example teaching content focused on the preliminary integration of interdisciplinary activities, and the combination of subject knowledge and Scratch was not emphasized enough (Figure 5).

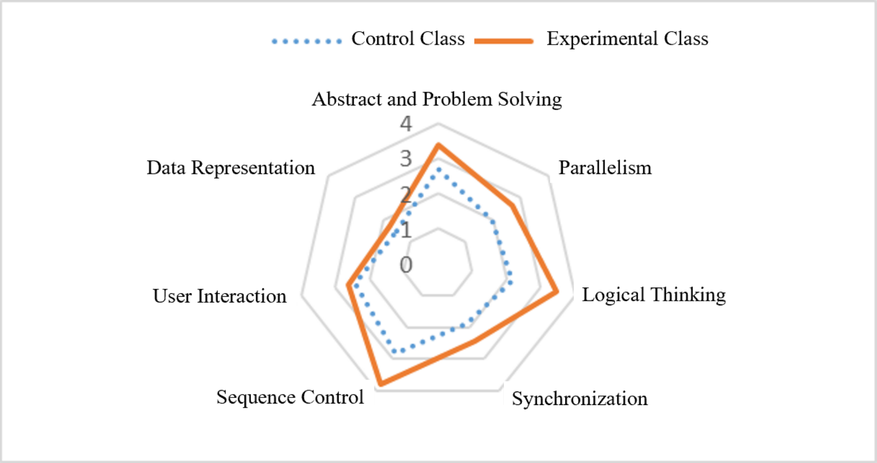


**Figure 5.** The results of post-test data on CT practices.

## 5. Conclusions and Discussion

### *5.1. Teaching Suggestions*

#### 5.1.1. Enhance the Practices in Existing Models or Strategies, and Enrich Research on the Development of CT

In-depth research into CT can be carried out for different audience groups, occupations, and classroom environments, as well as to explore the impacts of CT on daily learning and life [20]. Deepening the horizontal and vertical comparative study of existing models and strategies would be conducive to the further development of CT education. In terms of CT training activities, teachers need to choose the appropriate training modes according to the needs of their students, determine the teaching content and the teaching organization form, scientifically design teaching and learning activities, and then deliver CT education.

#### 5.1.2. Improve the Educational Resources for CT and Promote the Collaborative Development of Disciplines

According to collaborative theory, as a universal basic skill that can develop students' problem-solving and creativity, CT is the key to promoting the collaborative development of disciplines [39]. STEAM, interdisciplinary design, digital stories, and visual programming have been proven to be important strategies for developing CT, but CT resource systems are not perfect. In the future, CT education teachers should seize opportunities from the development of STEAM, use programming courses to build interdisciplinary teaching resources for CT, construct programming courses, gradually explore the "block-based programming to text-based programming + subject teaching" teaching resource system, and promote the application and popularization of CT in K–12 education, and even in higher education.

#### 5.1.3. Build a Multimodal Evaluation Method to Support Personalized Learning and Accurate Teaching

Domestic research into CT assessment has mainly focused on two aspects: the development of CT assessment scales, such as the CT scale based on programming [40] and the self-efficacy scale based on programming [41]; and the reference and application of the existing evaluation tools. Taking 1015 Chinese middle school students as a research sample, Bai Xuemei et al. [42] conducted a localization study on the CT scale (CTS) developed by Korkmaz et al. Through their data analysis, they verified that the scale could also be used to measure the CT levels of K–12 students in China. It is not difficult to see that although there are scientific measurement tools and scales for the evaluation of CT, how to automatically record and collect dynamic process data in CT education is still a dilemma. In the future, we should explore the functions of technology, such as the coding of CT learning behavior, the development of CT teaching data acquisition and analysis systems, and the automatic collection and analysis of the CT behaviors of students. Examining these areas could help to improve the accuracy of teaching and lead to the development of personalized recommendations.

### *5.2. Conclusions*

The experimental group was taught using an interdisciplinary activity design that was oriented to CT, and the control group was taught using traditional programming teaching activities. From our statistical analysis of the independent sample *t*-test and the paired sample *t*-test, we found that there were no significant differences in CT knowledge between the experimental group and the control group before the experiment. After the experiment, both the interdisciplinary activity design and the traditional programming teaching activities had significant effects on promoting the students' CT performances, although the former had a greater impact than the latter. By further analyzing the post-test knowledge levels of these two groups, we found that the knowledge of the experimental group was significantly higher than that of the control group regarding the dimensions of orders, cycles, events, and conditions. This showed that the interdisciplinary activity design could better promote student learning in those knowledge dimensions.

Following a Dr. Scratch analysis of the students' completed works, the results showed that after the experiment, knowledge of the dimensions of abstract thinking, problem-solving, logical thinking, synchronization, parallelism, and sequence control was significantly higher in the experimental group than in the control group. This showed that the interdisciplinary activity design could better promote student learning in those knowledge dimensions; however, neither strategy significantly promoted knowledge of the dimensions of data representation or user interaction.

### *5.3. Discussion*

The research found that an interdisciplinary activity design based on the concept of DBL could significantly promote the CT of students, especially in the dimensions of abstract thinking, problem-solving, and logical thinking. A study by Mustafa Saritepeci in 2020 also proved that DBL activities had an "intermediate" effect size on the creativity and problem-solving dimensions [20]. At this point, our findings seemed to show that DBL could be of utmost importance in students' CT skills when they solve problems creatively. This correlated fairly well with the findings of Wing, who suggested that two of the key issues in the acquisition and development of CT skills are formulating solutions for technological problems or contexts and achieving creative processes, including presenting concrete products. Zhang Yi et al. [43] also performed a similar experiment and their results showed that DBL teaching in STEM courses could significantly improve the calculation thinking and problem-solving abilities of primary school students, and play a certain role in improving their logical thinking. Therefore, DBL teaching could significantly improve the creativity of pupils.

## **6. Limitations**

This research was an exploration and an attempt to apply interdisciplinary activities to the development of CT. However, some limitations should be noted. First, only two classes of students participated in this research because of the limitations of the implementation conditions, so the sample size was small, which could have affected the final results. Secondly, the integration of subject knowledge and Scratch into the interdisciplinary activity design was not emphasized enough. The specific teaching content for the experimental and control groups needed to be improved. Thirdly, the reasons for why our interdisciplinary activity design based on the concept of DBL could significantly promote students' abstract thinking, problem-solving, and logical thinking need further investigation and research.

**Author Contributions:** Conceptualization, D.W.; Formal analysis, L.L.; Methodology, D.W. and G.R.; Software, L.L. and J.L.; Writing—original draft, L.L. and J.L.; Writing—review & editing, D.W., J.L., S.L. and G.R. L.L. carried out the experiment, and analyzed and interpreted the student data regarding the CT performance and the practical skills of CT. All authors were contributors in writing the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no competing interests.

## References

1. Wing, J. Computational Thinking. *Commun. ACM* **2006**, *49*, 33–35. [CrossRef]
2. CSTA and ISTE. Computational Thinking in K–12 Education Leadership Toolkit. Available online: http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf (accessed on 10 April 2020).
3. Bocconi, S.; Chioccariello, A.; Dettori, G. *Developing Computational Thinking in Compulsory Education*; JRC Working Papers; Publications Office of the European Union: Luxembourg, 2016.
4. Xie, Z.-X.; Cao, Y.-L. Strategies and methods for developing Computational Thinking of information technology students in primary and secondary schools. *China Audio Vis. Educ.* **2015**, *11*, 116–120.
5. Guo, S.-C.; Zhou, R.; Deng, C.-M.; Di, C.-Y.; Zhou, Q.-G. Research on it classroom teaching based on App Inventor and Computational Thinking. *China Audio Vis. Educ.* **2014**, *3*, 91–96.
6. Brennan, K.; Resnick, M. New frameworks for studying and assessing the development of computational thinking. In Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada, 13–17 April 2012; pp. 1–25.
7. Liang, Y.Z.; Liu, R.X.;Ren, L.L. Research on the precise teaching mode of human-computer cooperation oriented to the cultivation of computational thinking—Take the sixth grade information technology course "The Silk Road" as an example. *Mod. Educ. Technol.* **2022**, *3*, 51–60.
8. Burke, Q. The Markings of a New Pencil: Introducing Programming as-Writing in the Middle School Classroom. *J. Media Lit. Educ.* **2012**, *2*, 121–135. [CrossRef]
9. Zhou, P.H.; Niu, Y.K.; Wang, K.; Zhang, Y.; Li, X.; Shang, V.W. STEM engineering design teaching mode and application oriented to computational thinking training. *Mod. Distance Educ. Res.* **2022**, *1*, 104–112.
10. Kim, S.H.; Han, S.G. Design-Based Learning for Computational Thinking. *J. Korean Assoc. Inf. Educ.* **2012**, *16*, 319–326.
11. Jun, S.; Han, S.; Kim, S. Effect of design-based learning on improving computational thinking. *Behav. Inf. Technol.* **2016**, *36*, 43–53. [CrossRef]
12. Zhao, H.-C.; Zhang, N.-Y.; Ma, J.-W. STEM education interdisciplinary learning community: Promoting the change of learning style. *Open Educ. Res.* **2020**, *3*, 91–98.
13. Wang, X.-Q. Research on the development and evaluation of Computational Thinking in foreign primary and secondary schools for three-dimensional goals. *Res. Audio Vis. Educ.* **2014**, *7*, 48–53.
14. Fraillon, J.; Ainley, J.; Schulz, W. *IEA International Computer and Information Literacy Study 2018 ASSESSMENT FRAMEWORK*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 1–77.
15. Basu, S.; Biswas, G.; Sengupta, P.; Dickes, A.; Kinnebrew, J.S.; Clark, D. Identifying middle school students' challenges in computational thinking-based science learning. *Res. Pract. Technol. Enhanc. Learn.* **2016**, *11*, 1–35. [CrossRef] [PubMed]
16. Weintrop, D.; Beheshti, E.; Horn, M. Defining Computational Thinking for Mathematics and Science Classrooms. *J. Sci. Educ. Technol.* **2015**, *10*, 1–21.
17. Hsu, T.C.; Chang, S.C.; Huang, Y.M. How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Comput. Educ.* **2018**, *126*, 296–310. [CrossRef]
18. Farris, A.V.; Sengupta, P. Democratizing Children's Computation: Learning Computational Science as Aesthetic Experience. *Educ. Theory* **2016**, *66*, 279–296. [CrossRef]
19. Soleimania, A.; Herrob, D.; Greenc, K.E. Cyber PLAYce—A tangible, interactive learning tool fostering children's Computational Thinking through storytelling. *Int. J. Child-Comput. Interact.* **2019**, *20*, 9–23. [CrossRef]
20. Mustafa, S. Developing Computational Thinking Skills of High School Students: Design-Based Learning Activities and Programming Tasks. *Asia-Pac. Educ. Res.* **2019**, *1*, 35–54.
21. Zhang, Y.; Wang, J.; Zhang, J.; Zhu, Y.-H.; Zhou, W.-Q.; Wang, Y.-Y. Research on the development of primary school students' Computational Thinking by DBL teaching in STEM curriculum. *Audio Vis. Educ. Res.* **2020**, *5*, 81–88.
22. Grover, S.; Pea, R. Computational Thinking in K-12 A Review of the State of the Field. *Educ. Res.* **2013**, *1*, 38–43. [CrossRef]
23. Wing, J. Research Notebook: Computational Thinking—What and Why? *Link Mag.* **2011**, *6*, 20–23.

24. Lye, S.Y.; Koh, J.H.L. Review on Teaching and Learning of Computa tional Thinking through Programming: What is Next for K-12? *Comput. Hum. Behav.* **2014**, *41*, 51–61. [CrossRef]
25. Berl, M.; Wilensky, U. Comparing Virtual and Physical Robotics En vironments for Supporting Complex Systems and Computational Think ing. *J. Sci. Educ. Technol.* **2015**, *5*, 628–647.
26. Miller, P. Learning with a Missing Sense: What Can We Learn from the Interaction of a Deaf Child with a Turtle? *Am. Nals Deaf.* **2009**, *1*, 71–82. [CrossRef] [PubMed]
27. Lee, Y.J. Developing Computer Programming Concepts and Skills via Technology-enriched Language-art Projects: A Case Study. *J. Educ. Multimed. Hypermedia* **2010**, *3*, 307–326.
28. Sengupta, P.; Kinnebrew, J.S.; Satabdi, B. Integrating Computational Thinking with K-12 Science Education using Agent-based Compu tation: A Theoretical Framework. *Educ. Inf. Tech Nologies* **2013**, *2*, 351–380. [CrossRef]
29. Yang, Y.T.C.; Chang, C.H. Empowering Students through Digital Game Authorship:Enhancing Concentration, Critical Thinking, and Academic Achievement. *Comput. Educ.* **2013**, *68*, 334–344. [CrossRef]
30. Fessakis, G.; Gouli, E.; Mavroudi, E. Problem Solving by 5-6 Years Old Kindergarten Children in a Computer Programming Environment: A Case Study. *Comput. Educ.* **2013**, *63*, 87–97. [CrossRef]
31. Snodgrass, M.R.; Israel, M.; Reese, G.C. Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Comput. Educ.* **2016**, *9*, 1–17. [CrossRef]
32. Matsumoto, P.S.; Cao, J. The Development of Computational Thinking in a High School Chemistry Course. *J. Chem. Educ.* **2017**, *9*, 1217–1224. [CrossRef]
33. Zhu, K.; Jia, X.-X. Research on the development strategy of Computational Thinking Ability from the perspective of STEM. *Mod. Educ. Technol.* **2018**, *12*, 115–121.
34. Heintz, F.; Mannila, L.; Farnqvist, T. A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education. In Proceedings of the 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 12–15 October 2016; pp. 1–9.
35. Ismail, A.R. Computational Thinking to be Integrated into Primary and Secondary Curriculum. Available online: http://www.hardwarezone.com.my/tech-news-computationalthinking-be-integratedprimary-and-secondary-curriculum (accessed on 5 August 2019).
36. Kolodner, J.L.; Paul, J.C.; David, C. Problem-based learning meets case-based reasoning in the middle-school science classroom. *J. Learn. Sci.* **2003**, *4*, 495–547.
37. Doppelt, Y.; Mehalik, M.M.; Schunn, C.D. Engagement and Achievements: A Case Study of Design-Based Learning in a Science Context. *J. Technol. Educ.* **2008**, *2*, 18.
38. Marcos, R.G.; Juan-carlos, P.G.; Carmen, J.F Which cognitive abilities underlie Computational Thinking? Criterion validity of the Computational Thinking Test. *Comput. Hum. Behav.* **2016**, *9*, 678–691.
39. Yu, Y.; Xie, S.-X. Enabling computing thinking for education in the age of intelligence: Interpretation and Enlightenment of ISTE Computational Thinking Ability Standard (educators). *J. Distance Educ.* **2020**, *3*, 38–46.
40. Chu, Y.K.; Liang, J.C.; Tsai, M.J. Development of a Computational Thinking Scale for Programming. In *Proceedings of the International Conference on Computational Thinking Education 2019*; The Education University of Hong Kong: Hong Kong, China, 2020.
41. Kong, S.C. Development and validation of a programming self-efficacy scale for senior primary school learners. In *Conference Proceedings of the International Conference on Computational Thinking Education 2017*; The Education University of Hong Kong: Hong Kong, China, 2017; pp. 97–102.
42. Bai, X.-M.; Gu, X.-Q. Construction and application of evaluation tools for K12 students' Computational Thinking. *China Audio Vis. Educ.* **2019**, *10*, 83–90.
43. Zhang, Y.; Wang, J.; Zhang, L.; Zhu, Y.; Zhou, W.; Wang, Y.-Y. Research on DBL Teaching in STEM Curriculum to Cultivate Pupils' Computational Thinking. *Res. Audio Vis. Educ.* **2020**, *5*, 81–88.