Widyagogik, p ISSN 2303-307X, e ISSN 2541-5468

PRECODING ACTIVITIES TO IMPROVE STUDENT'S COMPUTATIONAL THINKING SKILLS

Ni Made Dwijayani *

*Program Studi Sistem Informasi, Institut Teknologi dan Bisnis STIKOM Bali, Denpasar, indonesia

Nimade_dwijayani@stikom-bali.ac.id , *https://orcid.org/0000-0002-7408-2948*

DOI: https://doi.org/10.21107/Widyagogik/v10i1.15671

Received May 22, 2022; June 12, 2022; Accepted July 12, 2022

Abstract

Coding means telling the computer to do something. In order to achieve this, the coder must understand what the problem is and what the best solution is. One of the things needed is pre-coding skills, including computational thinking skills. This study aims to improve students' computational thinking skills as a pre-coding ability. This classroom action research was conducted at a public elementary school in Dauh Puri, Denpasar, involving 20 first grade students. Data were collected based on observations in working on activity sheets and the results of student work on activity sheets. The data that has been collected was analyzed by quantitative descriptive. The results of data analysis showed that the students' computational thinking ability increased from pre-cycle activities, cycle I and cycle II respectively from 47.70 % to 65.05% and increased again to 81.50%. This increase in scores indicates that students' computational thinking skills can be improved by using student age-appropriate activity sheets as the basis for coding skills without having to involve a computer.

Keywords – Computational thinking; Precoding; Coding; Elementary school.

1. Introduction

Coding is the process of telling a computer what to do and how to accomplish it in accordance with pre-existing algorithms (McLennan 2017; Vorderman 2014). An algorithm is a set of instructions that must be followed in order for a task to be completed; it is a clear and specific series of activities that must be followed in order to solve the problem (Futschek & Moschitz, 2010; Mittermeir, 2013; Voronina et al., 2016). This mode of thinking in coding necessitates adopting a logical approach to organizing and evaluating data, followed by issue solutions by breaking it down into little, manageable bits. In other words, coding is a practical method of teaching mathematical reasoning (Futschek & Moschitz, 2010; Pugnali et al., 2017; Sullivan et al., 2017; Voronina et al., 2016). Students must first decide what they want the computer to tell them before they can consider coding. This type of problem-solving thinking is a computational thinking talent. This enables us to take a difficult situation, comprehend it, and build solutions. So computational thinking is a stage before kids actually code, and some refer to it as precoding abilities to make it easier for parents to comprehend. Figure 1 depicts the connection between computational thinking and coding.

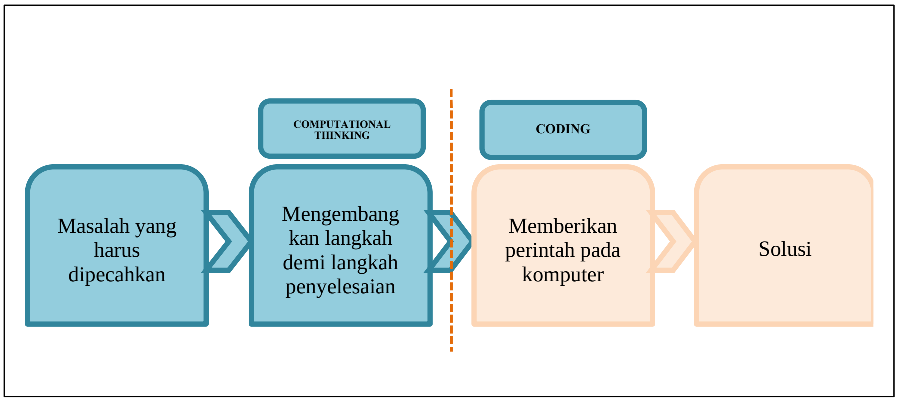

Figure 1. Connection between computational thinking and coding

Seymour Papert was the first to introduce computational thinking. Following that, further study was conducted in order to develop and explore the notion. Computational thinking is fundamentally linked to computer science. However, computational thinking is a method of thinking about issue solving and solution generation (Wing, 2006). (Hsu et al., 2018) conducted studies to identify 19 phases of Computational Thinking, including pattern detection, algorithm creation, and simulation.

Computational thinking may also be seen from the standpoint of issue solving, as one way to problem solving (Wing, 2008). Similarly, (Garca-Pealvo & Mendes, 2018) define computational thinking as an active problem-solving strategy in which students collect and evaluate data and construct solutions to issues using a set of ideas such as abstractions, patterns, and so on. Furthermore, computational thinking (as a thinking process) includes articulating issues so that they may be represented as computing phases and algorithms (Aho, 2012).

Computational thinking is described as a mental process that abstracts issues and formulates automated solutions (Yadav et al., 2014). (Barr & Stephenson, 2011) defines compulsive thinking as a problem-solving process with the following characteristics: problem formulation that allows the use of computers and other tools to help solve problems, data organization and analysis, data presentation through abstraction, automation of solutions by thinking algorithmically, analyzing solutions to obtain the most efficient solution, and generalization of this problem-solving process to a broader range of problems.

As a result, computational thinking is linked to computer science and how computer scientists think and reason. Computational thinking is also a problem-solving method. Then it is possible to infer that computational thinking is essentially a set of problem-solving procedures carried out by computer scientists using certain approaches.

Pattern is one of the teachings that is directly associated to computational thinking abilities. Students must recognize or identify particular patterns or rules from a row of pictures, shapes, or colors in order to study. Students will be able to break down the problem into easier-to-solve sections as a consequence of detecting patterns. Students are also asked to create patterns from difficulties.

Students must also be able to demonstrate an understanding of an issue and its solution. Students with this competence will find it simpler to generalize answers to many challenges. An algorithmic thinking process is also required for pattern recognition and pattern building.

Furthermore, the element of computational thinking ability used in this study relates to the component of computational ability defined by (Lee et al., 2012) and (Hsu et al., 2018). Understanding the problem, pattern identification, algorithmic reasoning, and selecting the best solution are the components.

2. Method

The Kemmis Taggart model (Febriyanti et al., 2021) is used in this study, with two cycles that involve planning, action and observation, and reflection. The study lasted two months in the even semester of 2021/2022 and concluded with an improvement in the quality of the process and learning results of children's problem solving skills. Twenty fifth-grade pupils from a public elementary school in Dauh Puri, Denpasar, participated in the study.

One of the most important parts of early childhood cognitive development is computational thinking skills. Computational thinking skills have distinct properties based on their developmental stage. Problem understanding, pattern identification, algorithmic thinking, and selecting the optimal solution are the tools used to assess this talent.

Table 1. The researcher assessed the optimal age for each ability and devised activities to facilitate learning.

Categories	Activities
Understanding the problem	• Recognize direction (right, left, forward, backward) • Sorting numbers (1-10)
Pattern recognition	• Organize things by size, shape, and color. • Capable of identifying, expanding on, and creating basic patterns

Algorithmic thinking	• Concentrate on the order of the activity steps.
Choose the best solution	• Recognize that there may be several solutions to a problem. • Recognize what a error is.

Researchers worked with classroom teachers to view pictures and video evidence of kids participating in class activities. In addition, researchers observed and interviewed children while they completed exercises using activity sheets created by researchers and teachers.

Before engaging in class activities, the teacher teaches how or the rules of "playing" (a term agreed upon by the researcher and teacher so that students do not feel as objects being observed). After students have mastered the game's rules, the teacher will assign a worksheet to assess students' computational thinking ability. Teachers and researchers examined pupils with weak computational thinking abilities (B), who began to develop (M) after being stimulated to grow in accordance with expectations (H), and who progressed extremely well (S).

Data analysis

Using activity sheets, researchers examined the results of children's growth in computational thinking skills. The researchers analyzed both qualitative and quantitative data. Observation of photo/video documentation of children's activities and interviews yielded qualitative data analysis. Using quantitative data analysis, the percentage increase in the achievement scale in the development of children's computational thinking abilities before and after participating in play activities was measured.

3. Result and Discussion

Results

Data collected earlier to the cycle through observations, interviews, and documentation. The preliminary assessment was completed in February 2022.

During the pre-cycle activities, researchers discovered issues with pupils' limited computational thinking abilities. The lack of variation in the exercises chosen by the instructor is one of the causes of children's inferior computational thinking abilities. Teachers frequently assign activities to perform tasks in books, which makes students less interested in engaging in the learning process.

The first cycle of learning was implemented in March 2022. According to the findings of the researcher's observations, there were still some children who needed to be led to play when they encountered hurdles while participating in recreational activities. Based on the findings of observations and interviews, the researcher created field notes that were utilized as evaluation materials. According to the data analysis results (see table 2), it attained a percentage of 65.05 percent in the first cycle and grew to 81.50 percent in the second cycle. This demonstrates that the computational thinking abilities of youngsters have improved. The results in table 2 show that playing with media based on computational thinking indicators may be utilized to improve the computational thinking skills of first grade primary school pupils.

Students are able to grasp what is known and asked when participating in play activities, recognize a pattern, identify the features of the problem and be able to apply various solutions discovered, and construct a pattern based on a problem. This investigation discovered several roadblocks, particularly in the third criteria. Despite the challenges, the findings of this study show that the computational thinking indicators for grade I students can be utilized to increase students' computational thinking abilities. This is demonstrated by the four criteria for playing activities used by researchers, as indicated in Table 2.

Tabel 2. Score of students' computational thinking skills

No	Students	Pre cycle	Cycle 1	Cycle 2
1	NO	57	71	81
2	IA	53	69	84
3	IS	41	70	83
4	ID	52	58	65
5	AP	41	80	91

6	AH	41	58	83
7	IS	50	58	81
8	NM	47	64	86
9	NY	50	59	85
10	NW	53	60	67
11	NP	41	58	80
12	IG	44	81	90
13	SH	44	83	91
14	WK	43	63	92
15	IP	56	59	64
16	AA	47	62	72
17	AS	53	63	81
18	WY	53	57	83
19	PY	42	61	86
20	PB	46	67	85

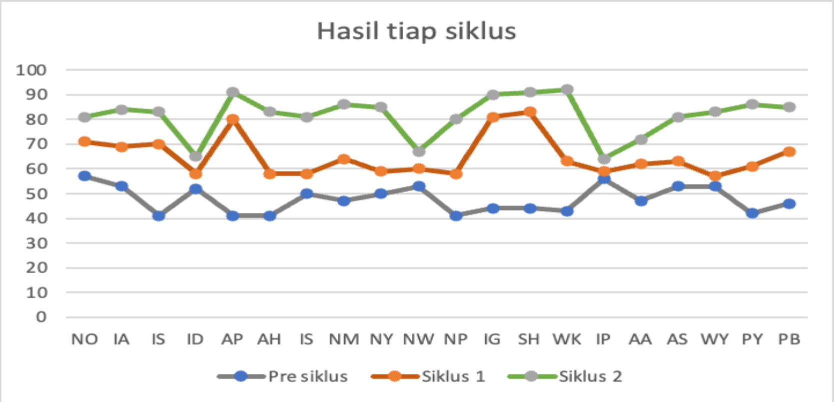

Figure 2. Observation score results

Figure 2 depicts the findings of observations made by researchers for computational thinking skills beginning with the pre-cycle, cycle I, and cycle II, with the first cycle outperforming the pre-cycle and the second cycle outperforming the first cycle.

Data Comprehension (Problem Understanding)

At the problem comprehension stage, all subjects have essentially identified and understood what is known and asked from the issue or problem. This is evident from the subject's responses and the interviews that were done. The subject can offer a solid explanation so that the subject does not have difficulty determining the direction and sorting the numbers when engaging in activities. The problem understanding stage is critical since the outcomes of this phase will be utilized to find a solution to the main challenge. Figure 3 depicts student effort in determining direction and sorting numbers.

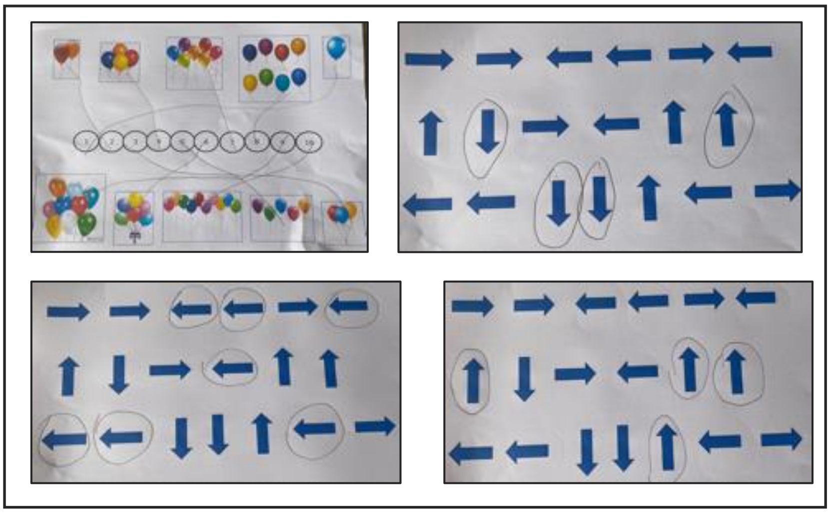

Figure 3. Examples of student work in determining direction and sorting numbers

Pattern Recognition (Pattern Recognition)

Figure 4 depicts subject replies relating to pattern recognition. Students must discover the link between the previous and next color to establish the pattern of the supplied image. There was still some uncertainty about the following color pattern in the pre-cycle and first-cycle stages. Students still require instructor direction during this stage. In cycle two, however, nearly all students were able to describe how they discovered the pattern of the provided hue and continued to fill in the blank pattern. As a result, all students have identified that the color pattern on the activity sheet is green at the top, followed by yellow, red, blue, and orange in a clockwise manner. The subject's response is essentially the same. The only variation is how students explain how they obtained the color pattern.

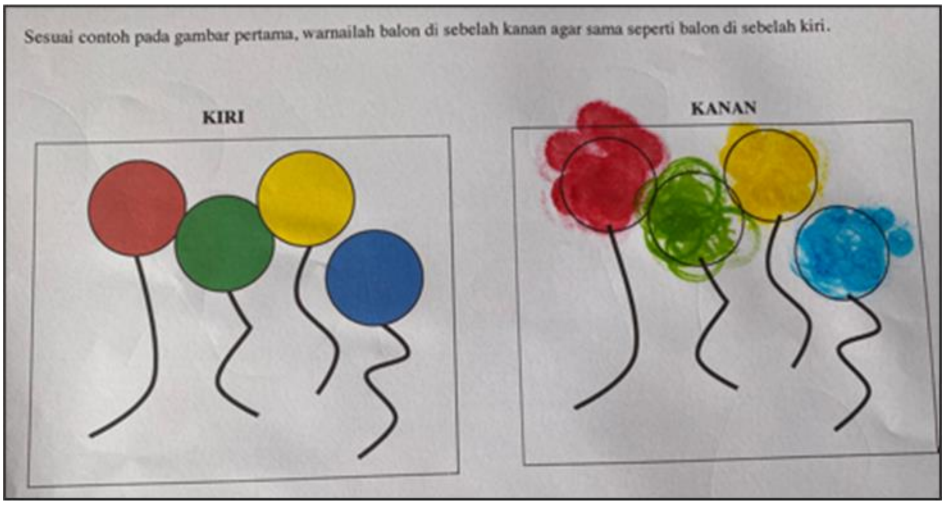

Figure 4. Examples of student answers about pattern recognition

Algorithmic Thinking (Thinking According to Algorithms)

Students are required to be able to recognize the features of the problem and use alternative solutions discovered for similar new challenges in this component. According to the findings of the analysis of the respondents' replies and interviews, 8 out of 20 individuals did not match these requirements in cycle II. The four individuals were unable to adapt the solution strategy they presented to additional tasks involving identifying the next picture based on the prior image. According to the interview findings, the four individuals had nearly identical replies, meaning that the idea employed to address this problem was different from the prior concept, which caused them difficulty. It can be noticed here that the four participants did not employ their computational thinking skills because they just remembered concepts rather than understanding them in the previous stage. The topic thinks that the new inquiry pertains to content that the teacher has not previously presented. However, if the subject knows the substance of the pattern in general, he can still grasp the meaning of the pattern. These results show that thinking abilities based on algorithms still need to be developed. Figure 5 shows some samples of student responses

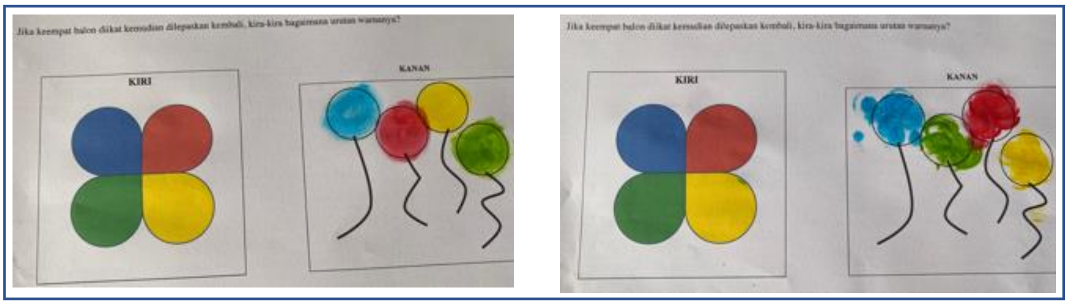

Figure 5. Examples of student answers related to algorithmic thinking

Solution Optimization (Choosing the Best Solution)

At this point, the subject is required to recognize that there are several solutions to a problem, and their responsibility is to select the best answer. Furthermore, in this stage, students learn about mistakes, which occur when a solution found does not address the problem. According to the findings of the respondents' responses and interviews, only three individuals did not grasp this level. Figure 6 is an example of a student's response.

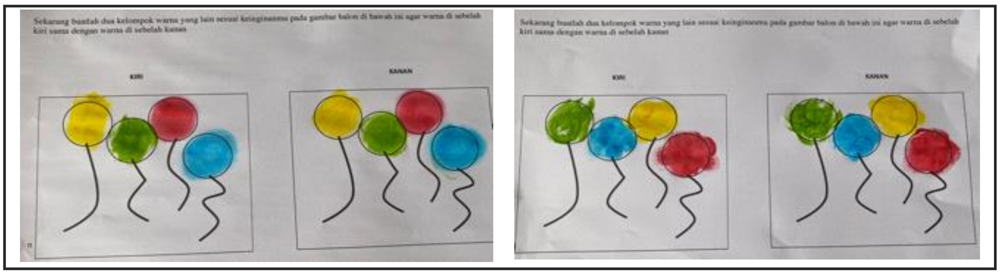

Figure 6. Examples of student answers regarding the best solution

Discussion

Several key conclusions concerning students' computational thinking abilities were gained as a result of these studies. In general, pupils already have high abilities on markers of issue comprehension, such as interpreting directions and sorting numbers from 1 to 10. Students exhibit high skills in the pattern recognition component as well, but only in the first sign of the algorithmic thinking component. Meanwhile, kids continue to fall short in other areas. These sections represent the elements of algorithmic reasoning and selecting the optimal answer.

Data Comprehension (Problem Understanding)

Several key insights about students' computational thinking abilities were produced based on these findings. In general, pupils already have high abilities on markers of problem comprehension, such as interpreting directions and sorting numbers from 1 to 10. Students also exhibit strong skills in the pattern recognition component, but only in the first sign of the algorithmic thinking component. Meanwhile, kids continue to struggle in other areas. These parts are indicative of the components of algorithmic thinking and selecting the optimal answer.

Understanding data becomes highly crucial in issue solving since it is the initial step in problem solution (P. J. Rich et al., 2019). Students are needed to know the facts that are known and asked from the questions at this stage, as well as to be able to define the problem. Unraveling is a process that breaks down a large difficulty into smaller difficulties that can be solved (Palts & Pedaste, 2020). When dealing with enormous difficulties and/or complicated jobs, decomposition is required (Selby and Woollard, 2013). As a result, these talents may be applied to daily tasks and become skills that help students live in society.

These findings suggest that issue comprehension abilities are the most challenging computational thinking skills to master. According to Selby (2013), the root of the difficulty is a shortage of practice questions completed by students, hence pupils are not accustomed to completing increasingly challenging issues. Furthermore, the questions or questions asked in class are frequently regular questions or routine inquiries. Teachers should ask non-routine questions because it is recognized that they boost pupils' problem-solving ability (Arslan & Altun, 2007).

When confronted with regular difficulties, we can directly forecast the process of understanding the problem and finding a solution. As a result, it merely offers pupils repetitious problem-solving assignments. Students do not generate additional ideas when addressing difficulties. Non-routine challenges are those with consequences that we cannot foresee in advance (Saygl, 2017). In other words, we cannot immediately address the problem. We can't address issues with tried-and-true procedures and formulae. We must occasionally employ a technique or formula. To solve it, we must do analysis, try and error, and think creatively. Furthermore, while tackling non-routine situations, thinking and approach are more crucial than getting solutions (Mayer et al., 1995). As a result, including non-routine challenges might generate better ideas for problem solutions.

Pattern Recognition (Pattern Recognition)

The subject recognizes patterns effectively in the pattern recognition component, although it should be noted that each subject has a varied degree of analyzing patterns. The course, for example, discusses how to identify the next color by knowing the preceding color. Another subject showed how to find the next color by knowing the previous color and how to determine the color at the nth place by knowing the previous color. In other words, there is a subject that explains the facts in the pattern further.

Algorithmic Thinking (Thinking According to Algorithms)

The third component of computational thinking, algorithmic thinking, is likewise unrelated to subject expertise. The researcher discovered that some respondents were unable to recognize the features of the supplied problem and could not apply the alternative answers gained for similar new challenges. The issue here is with the idea of color patterns and forms. Subjects still do not fully comprehend the meaning and structure of patterns in general. Because pupils lack a profound comprehension of meaning and structure, they are unable to answer additional issues that are identical to the prior ones. As a result, this ability must be developed. This is consistent with what has been said (Rijke et al., 2018) and (K. M. Rich et al., 2019). This is due to pupils' habit of memorizing formulae without understanding their significance. Understanding the meaning of formulae is crucial because it helps students remember formulas and utilize them more effectively. Furthermore, students can quickly determine whether the formula can be applied to a specific problem. To reduce these challenges in abstraction and generalization, teachers might assign tasks that do not enable students to utilize formulae directly, and students must consider the appropriateness of the formulas and problems assigned.

Solution Optimization (Choosing the Best Solution)

The last component of computational thinking in this research is the component of picking the optimal answer. The subject has been able to organize certain color patterns according to the problem, and the subject may build a methodical solution. At this point, pupils learn that the right color pattern may be observed from two directions: clockwise and counterclockwise. There are other pupils that create patterns by gazing from top to bottom or right to left. When one of the paths they picked revealed a contradiction, the pupils got perplexed and sought clarification from their teacher. This is what pupils refer to as identifying and correcting their own mistakes.

Futschek and Moschitz (2010) discovered that creating algorithms is one of the most effective learning approaches, particularly for identifying the optimum answer. Students continue to struggle with this component. This is consistent with (Burton, 2010) and (Plerou, 2016). Students struggle to improve their capacity to pick the best answer to a situation. One of the probable explanations is that students frequently resort to the usage of formulae without fully comprehending the problems and formulas. Students must grasp the supplied problem as well as the formula utilized in order to understand what aspects of the problem allow them to use the formula. As a result, when faced with additional difficulties, students will employ the proper formula for the relevant problem. Not only for mathematical problem solving, but also for issue solving in everyday life, particularly in society.

According to the Industrial Evolution Era 5.0, computational thinking is well developed today. Furthermore, mankind is now dealing with the Covid-19 epidemic. This necessitates being attentive to fast changes. Computational thinking will be a powerful talent that will prepare pupils for life in the Industrial Evolution Era 5.0 and beyond. As a result, the absence of computational thinking abilities in students must be addressed, as computational thinking is one of the cognitive talents that must be taught in all sectors of education (K. M. Rich et al., 2019). Furthermore, regardless of the subject, a student may utilize computational thinking to extend his thinking beyond obvious solutions, which stimulates student initiative and invention (Sanford & Naidu, 2016).

4. Conclusion

Because this program employs numerous play activities to make youngsters motivated, creative in expressing ideas, and finding the best solutions when they encounter issues while playing, the usage of precoding activities has an influence on students' computational thinking skills. Data from pre-cycle, cycle 1, and cycle 2 demonstrate an improvement in computational thinking skills. The use of pre coding begins with the instructor developing numerous play activities that correspond to the components of computational thinking abilities such as detecting direction, sorting numbers, determining color patterns, and picture patterns/shape patterns. When a youngster discovers an issue, the instructor stimulates and motivates the child by asking the child to name and describe the problem's chronology. Students still need to enhance their algorithmic thinking abilities, which is one of the four components of computational thinking skills listed above.

References

Aho, A. v. (2012). Computation and computational thinking. *Computer Journal*, *55*(7). https://doi.org/10.1093/comjnl/bxs074

Arslan, C., & Altun, M. (2007). Learning To Solve Non-routine Mathematical Problems. *Elementary Education Online*, *6*(1).

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1). https://doi.org/10.1145/1929887.1929905

Burton, B. A. (2010). Encouraging algorithmic thinking without a computer. *Olympiads in Informatics*, *4*.

Febriyanti, S., Istihapsari, V., & Afriady, D. (2021). Pengaruh Model Pembelajaran Problem Based Learning Untuk Meningkatkan Keaktifan Dan Hasil Belajar Siswa Pada Pembelajaran Tematik Kelas V Sd Negeri Balecatur I Tahun Pelajaran 2020 / 2021. In *Prosiding Pendidikan Profesi guru*.

Futschek, G., & Moschitz, J. (2010). Developing algorithmic thinking by inventing and playing algorithms. *Constructionism 2010*.

García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. In *Computers in Human Behavior* (Vol. 80). https://doi.org/10.1016/j.chb.2017.12.005

Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers and Education*, *126*. https://doi.org/10.1016/j.compedu.2018.07.004

Lee, T. Y., Mauriello, M. L., Ingraham, J., Sopan, A., Ahn, J., & Bederson, B. B. (2012). CTArcade: learning computational thinking while training virtual characters through game play. *CHI '12 Extended Abstracts on Human Factors in Computing Systems*. https://doi.org/10.1145/2212776.2223794

Mayer, R. E., Sims, V., & Tajika, H. (1995). Brief Note: A Comparison of How Textbooks Teach Mathematical Problem Solving in Japan and the United States. *American Educational Research Journal*, *32*(2). https://doi.org/10.3102/00028312032002443

McLennan, D. P. (2017). Creating coding stories and games. Teaching Young Children, 10(3). 18-21Retrieved October 02, 2019 from https:// www.naeyc .org/resou rces/pubs/tyc/feb20 17/creat ingcoding-stories-and-games

Mittermeir, R. T. (2013). Algorithmics for Preschoolers—A Contradiction? *Creative Education*, *04*(09). https://doi.org/10.4236/ce.2013.49081

Palts, T., & Pedaste, M. (2020). A model for developing computational thinking skills. *Informatics in Education*, *19*(1). https://doi.org/10.15388/INFEDU.2020.06

Plerou, A. (2016). Algorithmic Thinking and Mathematical Learning Difficulties Classification. *American Journal of Applied Psychology*, *5*(5). https://doi.org/10.11648/j.ajap.20160505.11

Pugnali, A., Sullivan, A., & Bers, M. U. (2017). THE impact of user interface on young children's computational thinking. *Journal of Information Technology Education: Innovations in Practice*, *16*(1). https://doi.org/10.28945/3768

Rich, K. M., Yadav, A., & Zhu, M. (2019). Levels of abstraction in students' mathematics strategies: What can applying computer science ideas about abstraction bring to elementary mathematics ? *Journal of Computers in Mathematics and Science Teaching*, *38*(3).

Rich, P. J., Egan, G., & Ellsworth, J. (2019). A framework for decomposition in computational thinking. *Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE*. https://doi.org/10.1145/3304221.3319793

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational thinking in primary school: An examination of abstraction and decomposition in different age groups. *Informatics in Education*, *17*(1). https://doi.org/10.15388/infedu.2018.05

Sanford, J. F., & Naidu, J. T. (2016). Computational Thinking Concepts for Grade School. *Contemporary Issues in Education Research (CIER)*, *9*(1). https://doi.org/10.19030/cier.v9i1.9547

Saygılı, E. ; (2017). Examining The Problem Solving Skills and The Strategies Used by High School Students in Solving Non-routine Problems. In *E-International Journal of Educational Research* (Vol. 8, Issue 2).

Selby, C., & Woollard, J. (2013). *Computational thinking: The developing definition.* https://eprints.soton.ac.uk/356481

Sullivan, A. A., Umaschi Bers, M., & Mihm, C. (2017). Imagining, playing, and coding with kibo: Using robotics to foster computational thinking in young children. *Proceedings of International Conference on Computational Thinking Education*.

Vorderman, C. (2014). Computer coding for kids: A unique step-by-step visual guide, from binary code to building games. London: Dorling Kindersley Ltd.

Voronina, L. v., Sergeeva, N. N., & Utyumova, E. A. (2016). Development of Algorithm Skills in Preschool Children. *Procedia - Social and Behavioral Sciences*, *233*. https://doi.org/10.1016/j.sbspro.2016.10.176

Wing, J. M. (2006). Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3).

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881). https://doi.org/10.1098/rsta.2008.0118

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. T. (2014).

Computational thinking in elementary and secondary teacher education. *ACM Transactions on Computing Education*, *14*(1). https://doi.org/10.1145/2576872