Measuring the Computational Thinking Abilities and Surveying Freshman Students' Opinions on Teaching and Learning Styles

Virawan Amnouychokanant, King Mongkut's University, Thailand
Surapon Boonlue, King Mongkut's University, Thailand
Saranya Chuathong, King Mongkut's University, Thailand
Kuntida Thamwipat, King Mongkut's University, Thailand

The Southeast Asian Conference on Education 2020
Official Conference Proceedings

Abstract
Computational Thinking (CT) is a key skill in the 21st century that everyone needs, rather than just being a programming skill used only by programmers. To develop students' systematic thinking and analytical abilities, we should add computational thinking to them. A sample group consisted of 89 freshman students attending the Department of Educational Communications and Technology of King Mongkut's University of Technology Thonburi, Thailand. They completed a Computational Thinking Test (CTt) and a students' opinions survey. Reliabilities as internal consistency of the CTt, measured by Cronbach's Alfa is $\propto = 0.79$. This test is aimed at measuring the students' CT abilities. The CTt had 20 multiple choice items and consisted of four components: Decomposition, Pattern Recognition, Abstraction, and Algorithm Design. The results revealed both males and females have equally average scores; however, their scores were lower than that of the criterion; consequently, they should be cultivated CT. The results from the students' opinions survey indicated that most respondents liked working with friends and learning by doing; in addition, they also admired an instructor who always applied problems in real life to teach them and combined Face-to-Face and Online Learning. These results can be applied in future research related to instructional design based on students' opinions and preferences.

Keywords: Computational Thinking Test, Decomposition, Pattern Recognition, Abstraction, Algorithm Design, Students' opinions

iafor
The International Academic Forum
www.iafor.org

Introduction

Computational Thinking (CT) is one of the daily life skills that everyone needs rather than being a programing skill used only by computer scientists. Especially learning in the 21st century that focuses on learning skills and innovation (3R4C) (Wing, 2006). CT can be combined with various subjects, but many teachers are still using programming languages to teach it (Lye & Koh, 2014; Zhong, Wang, Chen, & Li, 2016). CT is not only applied in computer programming but can also be used in mathematics and biology to train students' logical concepts, CT, and problem-solving skills (Hsu, Chang, and Hung, 2018).

CT is made up of four components, including decomposition, pattern recognition, abstraction, and algorithm design. The problem that we found from the literature review, was few studies designed Computational Thinking Test (CTt) that focused on four components as mentioned above. Most CTt always measured using block-based programming (Topalli & Cagiltay, 2018; Marcelino et al., 2018; Basogain et al., 2018; Erol and Kurt, 2017; Kazimoglu et al., 2012; Baytak & Land, 2011). In this study, we designed CTt that measured four components (decomposition, pattern recognition, abstraction, and algorithm design). We designed CTt that focused on thinking process rather than programming skill. CT is not a set of concepts for programming; it is a way of thinking that is sharpened through practice. It can be stated that CT is explaining and interpreting the world as a complex of information processes. (Denning & Tedre, 2019). After designing and validation CTt, we surveyed freshman students' opinions on learning styles, characteristics of favorite teachers, and characteristics of unacceptable teachers for the benefit of future researches in using it as data for instructional design based on students' opinions and preferences.

1.1 CT definitions

CT is a set of problem-solving processes that relate to expressing problems and solutions in ways that a computer could execute (Wing, 2014). It involves the mental skills and practices for designing computations that get computers to work for us and interpreting the world as a complex of information processes (Denning & Tedre, 2019). We clarified definitions of CT, as shown in Table 1.

CT is …	CT is not …	Resource
• thinking at multiple levels of abstraction • a way of human thinking to solve problems • a combination of mathematics and engineering thinking that can apply with various subjects • a fundamental skill in daily life that everyone needs	• the development process of programming language • copying the computer's thinking mode • a skill that only applied in computer programming • a programming skill used only by computer scientists	Wing, 2006; Grover & Pea, 2013; Hsu, Chang, and Hung, 2018

Table 1: The definitions of CT

1.2 The components of computational thinking

Computational Thinking is made up of four components, including decomposition, pattern recognition, abstraction, and algorithm design. The details of each component can be explained as follows:

Decomposition is breaking down data, processes, or problems into smaller. This makes complex problems easier to solve and large systems easier to design. To give you an idea, if you would like to understand how the motorbike system works, you can separate it into parts, then observe and test the function of each component will be easier to understand than analyze from large complex systems (Kilpeläinen, 2010; Hsu, Chang, and Hung, 2018).

Pattern Recognition is observing patterns, trends, and regularities in data. To illustrate, each cat has common characteristics. They have eyes, tails, fur, like to eat fish and meow. These common characteristics are called "pattern" when we can describe a cat, we will explain the characteristics of other cats in the same style itself (Hsu, Chang, and Hung, 2018).

Abstraction is focusing on the important information and ignoring unnecessary details. Although each cat will be in common. But it has different characteristics eventually. For example, some have different eyes and fur colors, some have short tails, some have long tails, some have fluffy hair, and some have no hair. An abstraction will screen the characteristics of each cat because the irrelevant details do not help us to explain the elementary characteristics of each cat. The process of screening out the irrelevant and focusing on a model helps us to solve problems which are called a model. When we have an abstract idea, it will give us a clearer idea model (Grover & Pea, 2013; Hsu, Chang, and Hung, 2018).

Algorithm Design is creating an order series of instructions for solving similar problems or for performing a task. When we need to order a computer to perform certain tasks, we need to write a program to execute the respective steps. Planning for a computer to meet our needs is called "algorithmic thinking" How well a computer will perform depends on how the algorithm works. The algorithm design is also useful for calculation, data processing, and various automated systems (Grover & Pea, 2013; Hsu, Chang, and Hung, 2018).

1.3 Computational thinking assessment

The measurement of computational thinking focuses on a thinking process rather than memorization. Primary school students were evaluated from classroom activities and behavioral observations such as explaining how to create a task as a set of step-by-step instructions to measure the results of a simple algorithm (Bers, 2010; Jovanov, Stankov, Mihova, Ristov, & Gusev, 2016; Kwon, Kim, Shim, & Lee, 2012). While high school students and university students were evaluated using a test that has focused on systematic thinking and problem-solving skill. The concepts of Computer Programming are transferred through storytelling or free exercises available on Code.org. There is no method to evaluate the effectiveness of these approaches; therefore, their validity is still unclear (Kalelioğlu, 2015). From the literature review, most studies in the past focused on block-based programming. Werner, Denner, Campe, & Kawamoto (2012) tried to use Alice to measure the understanding of abstraction, conditional logic, algorithmic thinking and other CT concepts while several studies used Scratch as a tool to measure CT abilities (Resnick et al., 2009;

Maloney, Resnick, Rusk, Silverman, & Eastmong, 2010; Clark, Rogers, Spradling, & Pais, 2013).

2. Method

2.1 Participants

A sample group consisted of 89 freshman students (30.34% male and 69.66% female) attending the Department of Educational Communications and Technology of King Mongkut's University of Technology Thonburi, Thailand. All participants enrolled in the Innovation in Educational Technology and Mass Communication course.

2.2 Computational Thinking Test

Before designing the Computational Thinking Test (CTt), we studied the other CTt such as the Talent Search Computational Challenge of Bebras Organization and the Test for Measuring Basic Programming Abilities (Mühling, Ruf, & Hubwieser, 2015). We created a Computational Thinking Test with a length of 30 multiple choice items. After a content validation process through three experts' judgement, the final one was consisted of 20 items length. The CTt was built on the following principles:

- Aim: CTt aims to measure the students' CT abilities.
- Target population: CTt is specifically designed for students in higher education.
- Instrument type: multiple choice test with 4 answer options.
- Length and estimated completion time: 20 items; 30 mins.
- Computational concept addressed: each item addresses one or more of the following four computational thinking components (Decomposition, Pattern Recognition, Abstraction, and Algorithm Design). The example of CTt items translated into English is shown in Figures 1-2; with their details below.

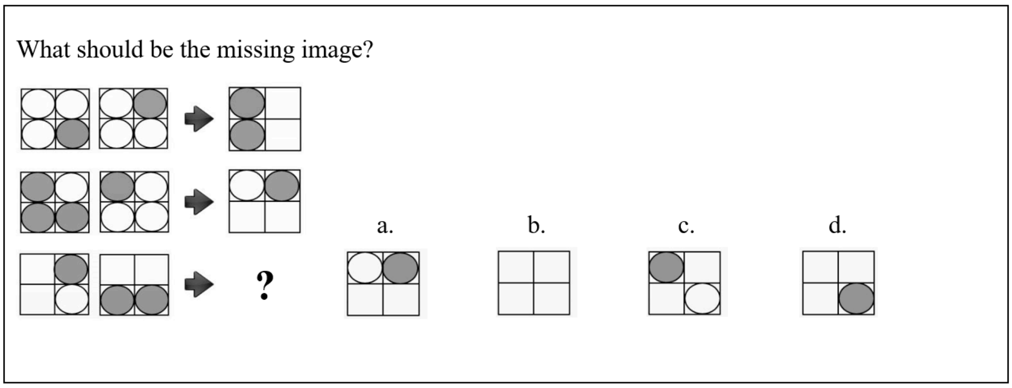
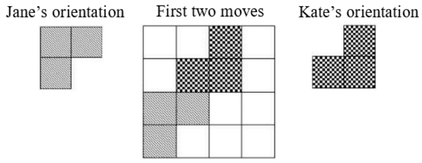


Figure 1: item 4; pattern recognition

Jane and Kate are playing L-Game on a 4x4 board. They take turns placing L-shaped pieces. Every piece placed by Jane and Kate is oriented as shown below and no two pieces overlap. Pieces cannot be moved after they are placed. A player loses the game when it is their turn, but it is not possible to place a piece according to the rules above.

An example where Jane goes first is shown below. In this example, Jane can win the game by placing a piece in the bottom-right corner.



Question: Jane has nine possible first moves.
In how many of them is she guaranteed to win
no matter how pieces are placed in following turns?

a. 1 b. 2 c. 3 d. 4

Figure 2: item 16; decomposition and abstraction

2.3 Procedure

After the validation process of CTt, we have used 20 questions to create an online test in the Google form for easy access to the samples. A sample of 89 people will be divided into two groups to do the tests in the computer room. The test-takers can use the computers in the computer room or use their mobile phones depending on the convenience of each person. During the test, two invigilators walked around to prevent cheating. After collecting data from the sample, we are going to analyze the data with SPSS software version 18.0.

3. Results and discussion

3.1 The qualities of CTt

Table 2 shows the validation of CTt. The Index of Item-Objective Congruence (IOC) along the 20 items is 0.87; that can be interpreted as good content validity (Sireci, 2007; Sireci, 1998). While Reliabilities as internal consistency of the CTt, measured by Cronbach's Alfa is 0.79 that can be considered as high reliabilities (Nunnally & Bernstein, 1994; Roman-Gonzalez, Perez-Gonzalez, & Jimenez-Fernandez, 2017). The average along the 20 items is p = 0.59 (medium difficulty); ranging from p = 0.26 (item 3; quite difficult) to p = 0.76 (item 11; quite easy).

Summarizing, it can be stated that the CTt has an appropriate degree of difficulty (medium) for the sample group and has characteristics of a good test (Cronbach & Thorndike, 1971; Messick, 1980). The qualities of CTt, as shown in Table 2.

The qualities	CTt value	Criterion
Index of Item-Objective Congruence (IOC)	0.87	IOC > 0.5
Cronbach's alpha (α) reliability coefficient	0.79	Ranging between 0 and 1.00 0.71-1.00; high reliability 0.41-0.70; medium reliability 0.21-0.40; low reliability 0.00-0.20; very low reliability
Difficulty (P)	0.59	P = 0.2 to 0.8; acceptable P > 0.80; too easy P < 0.2; too difficult

Table 2: The qualities of CTt

3.2 CT abilities

The results of the four components of CT abilities are shown in Table 3. The component that has the highest mean is abstraction and the component that has the lowest mean is algorithm design. Therefore, the students should be developed algorithm design, which is considered the most important component of the CT. This is because an algorithm can be defined as a step by step procedure for achieving any goals and it is used to find the best possible way of solving a problem, which makes it easy to understand for anyone even without programming knowledge (Hsu, Chang, & Hung, 2018; Barr & Stephenson, 2011; Grover & Pea, 2013).

From Table 4 indicates gender differences that affect CT abilities. There is no statistically significant difference between genders ($p > 0.01$), but their scores are bottom level of CT (Table 5). Therefore, they need to be developed computational thinking to have problem-solving skills in a systematic way and create something new (Günbatar, 2019; Hsu, Chang, & Hung, 2018; Lye & Koh, 2014).

CT Components	Gender	n	Mean	S.D.	t
Decomposition	Male	27	2.80	.698	-0.296
	Female	62	2.90	.751	
Pattern Recognition	Male	27	2.90	.903	-0.427
	Female	62	3.03	.800	
Abstraction	Male	27	4.72	.320	0.611
	Female	62	4.60	.371	
Algorithm Design	Male	27	1.92	.989	-0.620
	Female	62	2.15	.998	

Table 3: The results of each CT component

CT Components	Gender	n	Mean	S.D.	t	Sig
Overall	Male	27	11.86	1.423	-0.602	0.549
	Female	62	12.10	1.690		

Table 4: The results of overall CT components

Mean	Interpretation
18.00-20.00	Excellent
15.00-17.99	Good
12.00-14.99	Acceptable
10.00-11.99	Passed
below 10	Failed

Table 5: Criteria for CT abilities levels

3.3 Freshman students’ opinions on teaching and learning styles

After measuring the computational thinking abilities, we surveyed freshman students’ opinions on teaching and learning styles. According to Table 6, most respondents liked to be engaged with others, work on teams, and ask peers for feedback in order to learn and they preferred learning by doing and hands-on experience. In addition, we surveyed the students’ opinions of the characteristics of teachers they liked and disliked. Most students liked the teacher who applied problem in real life to teach them and combined face to face and online learning. On the contrary, most students disliked the teachers who unchanged their teaching styles and methods.

Hence, it can be assumed that the students who like collaborative learning, sharing their ideas, and carrying out projects together, tend to be enjoy with learning, more attentive and active which leads to the creation of effective learning, consistent with past studies (Chen, Li, & Chen, 2020; Hernández-Sellés, Muñoz-Carril, & González-Sanmamed; 2019).

Topic	Items	n	%
1. Learning styles	1.1 Prefer seeing the info and to visualize the relationships between ideas	22	13.7
	1.2 Prefer hearing info rather than reading it or seeing it displayed visually	16	9.9
	1.3 Prefer reading and writing rather than hearing or seeing images	12	7.5
	1.4 Prefer learning by doing and hands-on experience	47	29.2
	1.5 Like to be engaged with others, work on teams, and ask peers for feedback	45	27.9
	1.6 Prefer to be alone when learning something and study by yourself	19	11.8
2. Characteristics of favorite teachers	2.1 Set high expectations for all students	18	6.1
	2.2 Focus on shared decision-making and teamwork	37	12.5
	2.3 Apply problems in real life to teach	53	18.0
	2.4 Combine Face-to-Face and Online Learning	42	14.2
	2.5 Deep Knowledge of and Passion for the Subject Matter	31	10.5
	2.6 Encourage discussion in the classroom	27	9.2
	2.7 Excellent preparation and organization skills	23	7.8
	2.8 Excellent Communication Skills	29	9.8
	2.9 Using a variety of teaching style and innovative approaches	35	11.9
3. Characteristics of unacceptable teachers	3.1 Lack of classroom management	24	17.0
	3.2 Lack of content knowledge	30	21.3
	3.3 Lack of organizational skills	22	15.6
	3.4 Routine and unchanging in their teaching styles and methods	39	27.7
	3.5 Unable to diagnose learning problems	26	18.4

*Respondents could select more than one answer in each topic.

Table 6: Results of surveying freshman students' opinions on teaching and learning styles

3.4 Instructional design

We collected the teaching techniques that match the students' preferences, as shown in Table 7. There are many reasons why the student-centered approach is important. It addresses all the essential needs of students and places high emphasis on relevance and engagement of students. It provides the opportunity to foster collaborative learning. The instructor should design activities in such a way that students have to involve their peers in completing the tasks. This approach not only encourages collaboration but also fosters teamwork (Barak & Assal, 2018). Using games and challenges develops problem-solving skills, which is useful when students encounter similar problems in real life (Argaw et al., 2017).

Students' preferences	Teaching techniques	Advantages for students	References
Prefer learning by hands-on experience and apply problems in real life	• Project-based learning • Problem-based learning • Inquiry Method • Case study method	• Allows students to experiment with trial and error, learn from their mistakes, and understand the potential gaps between theory and practice	(Akınoğlu & Tandoğan, 2007; Abdelkhalek, Hussein, Gibbs, & Hamdy, 2010; Beaumont, Savin-Baden, Conradi, & Poulton, 2014; Argaw, Haile, Ayalew, & Kuma, 2017)
Like to be engaged with others, work on teams, and ask peers for feedback	• Project-based learning • Team Assisted Individualization (TAI) • Student Teams Achievement Divisions (STAD) • Team-Games-Tournaments (TGT) • Think pair share • Group Investigation • Team Word-Webbing • Discussion Method	• Enhances communication and social skills • Increases diversity of solutions and alternatives • Encourage students to find knowledge by themselves. • Allows students to work towards a common goal	(Barak & Assal, 2018; Crismond, 2011; Kolmos, 1996; Savery, 2006; Zadok & Voloch, 2018; Rogers, Cross, Gresalfi, Trauth-Nare, & Buck, 2011)

Combine Face-to-Face and Online Learning	Blended learning • Outside-In • Supplemental • Inside-Out • Flex • Lab Rotation • Station Rotation • Individual Rotation • Self-Directed • Project-Based • Remote • Flipped Classroom • Mastery-Based	• Enables students to learn at their own pace • Provides student autonomy • Incorporates a variety of instructional approaches	(Basogain, Olabe, Olabe, & Rico, 2018; Li, He, Yuan, Chen, & Sun, 2019; Yigsaw et al., 2019; Akçayır & Akçayır, 2018; AlJarrah, Thomas, & Shehab, 2018)

Table 7: The details of the teaching techniques that match the students' preferences

4. Limitations and future studies

The limitations of this study include its sample. This sample only includes freshman students from the Department of Educational Communications and Technology of King Mongkut's University of Technology Thonburi and is not a representative sample of Thailand and other regions.

Future studies may collect more data from other levels of various universities. Surveying students' opinions on teaching and learning styles may not adequate for the instructional design to improve CT of students. Future studies may collect data about teaching tools that can support in CT courses.

5. Conclusion

A Computational Thinking Test (CTt) was developed and validated. The CTt has an appropriate degree of difficulty (medium; p = 0.59) for the sample group and has characteristics of a good test (Cronbach & Thorndike, 1971; Messick, 1980). Four components of CT were measured in this test. A sample group consisted of 89 freshman students attending the Department of Educational Communications and Technology of King Mongkut's University of Technology Thonburi, Thailand. The results of freshman students' CT abilities showed that their scores are bottom level of CT. Therefore, they need to develop computational thinking in order to have problem-solving skills in a systematic way (Günbatar, 2019; Hsu, Chang, & Hung, 2018; Lye & Koh, 2014). After measuring the computational thinking abilities, we surveyed freshman students' opinions on teaching and learning styles. The results indicated that most students liked working with friends and learning by doing; besides, they also liked an instructor who applied problems in real life to teach them and combined face to face and online learning. These results can be applied in future research related to instructional design based on students' opinions and preferences. Examples of Teaching techniques that match the students' preferences such as Problem-based Learning, Project-based Learning, and Blended Learning, etc.

Acknowledgements

The authors would like to express sincere thanks to Petchra Pra Jom Klao Ph.D. Research Scholarship for supporting the expense of research.

References

Abdelkhalek N., Hussein A., Gibbs T., Hamdy H. (2010). Using team-based learning to prepare medical students for future problem-based learning. *Med Teach 32*(2), 123–129.

Akınoğlu, O., & Tandoğan, R.Ö. (2007). The effects of problem-based active learning in science. education on students' academic achievement, attitude and concept learning. *Eurasia Journal of Mathematics, Science and Technology Education, 3*(1), 71–81.

Akçayır, G., & Akçayır, M. (2018). The flipped classroom: A review of its advantages and challenges. *Computers & Education, 126*, 334–345.

AlJarrah, A., Thomas, M. K., & Shehab, M. (2018). Investigating temporal access in a flipped classroom: Procrastination persists. *International Journal of Educational Technology in Higher Education, 15*(1), 1.

Almanasreh, E., Moles, R., & Chen, T. F. (2019). Evaluation of methods used for estimating content validity.

Argaw, A. S., Haile, B. B., Ayalew, B. T., & Kuma, S. G. (2017). The effect of problem-based learning (PBL) instruction on students' motivation and problem-solving skills of physics. *Eurasia Journal of Mathematics, Science and Technology Education, 13*(3), 857–871.

Barak, M. & Assal, M. (2018). Robotics and STEM learning: students' achievements in assignments according to the P3 task taxonomy – practice, problem solving, and projects. *International Journal of Technology and Design Education, 28*(1), 121-144.

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), 48–54.

Basogain, X., Olabe, M. A., Olabe, J. C., & Rico, M. J. (2018). Computational Thinking in pre-university Blended Learning classrooms. *Computers in Human Behavior, 80*, 412-419.

Baytak, A., & Land, S. M. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom. *Education Tech Research Dev, 59*, 765-782.

Beaumont, C., Savin-Baden, M., Conradi, E., & Poulton, T. (2014). Evaluating a second life problem-based learning (PBL) demonstrator project: What can we learn? *Interactive Learning Environments, 22*(1), 125–141.

Bers, M. U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice, 12*(2), 2.

Chen, C.-M., Li, M.-C., & Chen, T.-C. (2020). A web-based collaborative reading annotation system with gamification mechanisms to improve reading performance. *Computers & Education, 144*.

Clark, J., Rogers, M. P., Spradling, C., & Pais, J. (2013). What, no canoes? Lessons learned while hosting a Scratch summer camp. *Journal of Computing Sciences in Colleges, 28*, 204-210.

Crismond, D. P. (2011). "Scaffolding strategies for integrating engineering design and scientific inquiry in project-based learning environments", in Barak, M. and Hacker, M. (Eds), *Fostering Human Development Through Engineering and Technology Education,* 235-255.

Cronbach, L. J., & Thorndike, R. L. (1971). Educational measurement. *Test validation*, 443-507.

Denning, P. J., & Tedre, M. (2019). *Computational Thinking*. United States: MIT Press Ltd.

Erol, O., & Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Computers in Human Behavior, 77*, 11-18.

Grover, S., & Pea, R. (2013). Computational thinking in K–12 a review of the state of the field. *Educational Researcher, 42*(1), 38–43.

Günbatar, M. S. (2019). Computational thinking within the context of professional life: Change in CT skill from the viewpoint of teachers. *Education and Information Technologies, 24*(5), 2629-2652.

Hernández-Sellés, N., Muñoz-Carril, P.-C., & González-Sanmamed, M. (2019). Computer-supported collaborative learning: An analysis of the relationship between interaction, emotional support and online collaborative tools. *Computers & Education, 138*, 1-12.

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 & Education, 126*, 299-300.

Jovanov, M., Stankov, E., Mihova, M., Ristov, S., & Gusev, M. (2016). Computing as a new compulsory subject in the Macedonian primary schools curriculum. *Global engineering education conference (EDUCON), 2016 IEEE* (pp. 680–685). IEEE.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. Org. *Computers in Human Behavior, 52*, 200–210.

Kazimoglu, C., Kiernan, M., Bacon, L., & MacKinnon, L. (2012). Learning Programming at the Computational Thinking Level via Digital Game-Play. *Procedia Computer Science, 9, 522-531.*

Kilpeläinen, P. (2010). Do all roads lead to Rome? (Or reductions for dummy travelers). *Computer Science Education, 20*(3), 181–199.

Kolmos, A. (1996), Reflections on project work and problem-based learning, *European Journal of Engineering Education, 21*(2), 141-148.

Kwon, D. Y., Kim, H. S., Shim, J. K., & Lee, W. G. (2012). Algorithmic bricks: a tangible robot programming tool for elementary school students. Education. *IEEE Transactions, 55*(4), 474-479.

Li, C., He, J., Yuan, C., Chen, B., & Sun, Z. (2019). The effects of blended learning on knowledge, skills, and satisfaction in nursing students: A meta-analysis. *Nurse Education Today, 82*, 51-57.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmong, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education, 10*(4), 1-15.

Marcelino, M. J., Pessoa, T., Vieira, C., Salvador, T., & Mendes, A. J. (2018). Learning Computational Thinking and scratch at distance. *Computers in Human Behavior, 80, 470-477.*

Messick, S. (1980). Test validity and the ethics of assessment. *American Psychologist, 35*(11), 1012-1027.

Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the 10th workshop in primary and secondary computing education,* 2-10.

Nunnally, J. C., & Bernstein, I. H. (1994). *Psychometric theory* (3rd ed.). New York: McGraw-Hill.

Resnick, M., Maloney, J., Hernandez, A. M., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: programming for all. *Communications of the ACM, 52*(11). Retrieved from http://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf.

Rogers, M. A. P., Cross, D. I., Gresalfi, M. S., Trauth-Nare, A. E. & Buck, G. A. (2011). First year implementation of a project-based learning approach: the need for addressing teachers' orientations in the era of reform, *International Journal of Science and Mathematics Education, 9*(4), 893-917.

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., & Jimenez-Fernandez, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678-691.

Savery, J. R. (2006). Overview of problem-based learning: definitions and distinctions, *Interdisciplinary Journal of Problem-Based Learning, 1*(1).

Sireci, S. G. (1998). The construct of content validity. *Social Indicators Research, 45*(1), 83-117.

Sireci, S. G. (2007). On validity theory and test validation. *Educational Researcher, 36*(8), 477-481.

Topalli, D., & Cagiltay, N. E. (2018). Improving programming skills in engineering education through problem-based game projects with Scratch. *Computers & Education, 120, 64-74.*

Wing, J. M. (2014). Computational Thinking Benefits Society. *40th Anniversary Blog of Social Issues in Computing.*

Yigsaw, M., Tebekaw, Y., Kim, Y.-M., Kols, A., Ayalew, F., & Eyassu, G. (2019). Comparing the effectiveness of a blended learning approach with a conventional learning approach for basic emergency obstetric and newborn care training in Ethiopia. *Midwifery, 78*, 42-49.

Zadok, Y. and Voloch, N. (2018). Applying PBL to teaching robotics, *International Journal of Innovation and Learning, 24*(2), 138-151.