

The concept of computational thinking toward information and communication technology learning

By asto buditjahjanto

PAPER • OPEN ACCESS

The concept of computational thinking toward information and communication technology learning

8

To cite this article: R Harimurti *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **535** 012004

8

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

The concept of computational thinking toward information and communication technology learning

R Harimurti¹, Ekohariadi², Munoto³ and IGP Asto B⁴

¹Informatics Department, Surabaya State University, Surabaya, Indonesia

^{2,3,4}Electrical Engineering Department, Surabaya State University, Surabaya, Indonesia

E-mail: rinaharimurti@unesa.ac.id

Abstract. Computational thinking is a process of problems solving and designing systems using concepts in computer science. Computational thinking is based on a computational principle that involves a wide range of approaches and skills with applications in several disciplines. Computational thinking has been widely applied in the curriculum of various levels of education while still basing on the principles and characteristics of a strong computational thinking. To train the computational thinking to students it must include the computational thinking material in the compulsory material. This material needs to be held since the level of basic education with a simple problem level, the level of the material will increase along with the increase of education level. This article discusses the implementation of computational thinking on information and communication technology learning using a simple programming language that is Scratch. This programming language will make it easier for students to understand the concept of programming by using blocks so that students will easily learn. As a basis for information and communication technology learning, a measurement scale has been developed to measure students' computational thinking ability. Computational thinking uses a five-point Likert scale consisting of 29 items from 5 developed indicators.

1. Introduction

Learning is an activity that involves a person in an effort to gain knowledge, skills, and positive values by utilizing various sources for learning. Learning can involve two parties, namely students as learners and teachers as facilitators or teachers. The most important thing in the learning activities is the learning process (learning process).

Current technological developments have made computer devices a very effective learning tool, as they are widely used for routine tasks, helping to solve complex problems as well as designing systems. The basic principle of problem-solving techniques and their application is the concept of development of computational thinking (CT).

Computational Thinking (CT) is essentially problem-solving, designing systems, and understanding human behaviour, drawing on fundamental concepts to computer science can be performed by humans or machines (computers) [1] CT has an important role in computer application development, but CT can also be used to support problem solving in all disciplines, including humanities, mathematics and science. Currently at all grade levels, CT competence has been applied as it is a basic principle of problem-solving techniques and their application. Programming is more than coding, as it brings students to computational thinking that involves solving problems using computer science concepts such as abstraction and decomposition [2] Even for non-computing majors,



Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Published under licence by IOP Publishing Ltd

CT is applicable and useful in everyday life. Not surprisingly, having such capability is a must for someone living in the digital age today.

This article will discuss how the conceptual framework of CT underlies the development of ICT learning. Section 2 discusses the methodology, and Section 3 is the results and discussion. Finally, Section 4 summarizes the research.

2. Method

Literature study is used in the compilation of this article, this method is done by collecting references, books and journals related to the discussion of theory, analysis, research that has been done and also the implementation of CT on ICT learning. The journal used as a source begins from 2006 to the latest in 2017. The journals used for reference can be grouped as follows: basic concept of CT, CT development, curriculum developed based on CT and CT implementation on various levels of education. Keywords in the search of journals that support the writing of this article is Computational Thinking (CT).

3. Result and Discussion

3.1 Basic Principles of Computational Thinking (CT)

Computational thinking (CT) is based on computational principle that is a term that covers a variety of conceptual and engineering tasks, so computational thinking involves a broad set of approaches and is with applications in multiple disciplines. Computational thinking (CT) developed by Jeannette Wing [1] to describe a set of thinking skills, habits and approaches that are integrated to solve complex problems using computers. CT developed several key principles [3] which are listed below. (1) how to solve problems and design the system refers to the basic concepts on computer science (2) create and use different levels of abstraction to understand and solve problems more effectively (3) think algorithms and with the ability to apply math concepts to be developed more efficiently (4) understand the consequences of scale, not only for efficiency reasons but also for economic and social reasons

According to Denning [4] the elements of computational thinking include the seven principles of calculation, communication, coordination, recollection, automation, evaluation, and design. These seven principles form a useful foundation for recognizing, organizing, and categorizing examples of computational thinking and building a framework that can translate computational thinking to contexts outside of computer science [5] Figure 1 shows the seven principles.

The seven principles mentioned above lead a meaning that CT is understandable by everyone [6], which means: (1) understanding which aspects of the problem are acceptable for computing, (2) evaluation of compatibility between tools and computing techniques and problems, (3) understanding the limitations and strengths of computing tools and techniques, (4) applying or adapting a computing tool or technique for new uses, (5) recognizing the opportunities to use computing in new ways (6) implementing computing strategies such as dividing and conquering in any domain.

3.2 Characteristics of Computational Thinking (CT)

As aforementioned, the article written by Wing [1] stated that CT has the following characteristics: (1) conceptualization, not programming (2) basic skills, not memorizing (3) the human way, not the computer (4) complete and combine mathematical thinking and techniques (5) ideas are not artefacts (6) for everyone everywhere (7) a scientific problem that is intellectually challenging and interesting (8) a person can master computer science and do anything

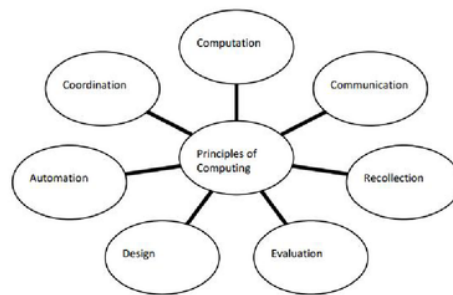


Figure 1. Seven principles of computational thinking.

CT is related to the ability to think recursively, using abstraction and decomposition when completing complex tasks. Some researchers focus on skills that are characteristic of CT. CT represents a cognitive process that accommodates logic, algorithmic, analytical, mathematical, technical and creative thinking [7]. Three dimensions of computational thinking are computational concepts, computational practices and computational perspectives. The characteristics of computational thinking are being able to [1]: (1) provide troubleshooting using a computer or other device (2) organize and analyse data (3) perform data representation through abstraction with a model or simulation (4) automate solutions through algorithmic thinking (5) identify, analyse and implement solutions with various combinations of steps / ways and resources that are efficient and effective. (6) generalize solutions for different problems.

Thinking computationally is a fundamental skill for everyone, not just computer scientists. This suggests that the importance of integrating computing ideas into other disciplines. The following are elements or elements of CT. (1) **Decomposition**: The ability to break down data, processes or problems (complex) into smaller parts or become manageable tasks. For example, breaking up 'Drive / Direction' in a computer based on its constituent components: Files and Directory. (2) **Pattern Recognition**: The ability to see the similarities or even differences in patterns, trends and regularities in the data that will later be used in making predictions and presenting the data. For example, recognize the pattern of document files, file systems, execution files or data structures / files. (3) **Abstraction**: Generalize and identify general principles that produce such patterns, trends and regularities. For example, by placing all system files in the Windows folder, program files in the Program Files folder, data files / documents in the MyDocument Folder and supporting files in a separate Drive / Direction. (4) **Algorithm**: Develop the same step-by-step troubleshooting instructions so that others can use that step / information to solve the same problem.

3.3 Implementation of CT in ICT learning

In most societies today, teacher organizations, accrediting bodies, and government agencies are actively promoting the teaching of computational thinking (CT) skills [8]. Discourse on the essence of CT continues to be done, integrated efforts have made computer science used as a requirement of high school graduation in many US states. Computational thinking (CT) is in the realm of educational innovation, as a part of problem-solving skills that students must acquire to thrive in a digital world full of software-driven objects. [9]

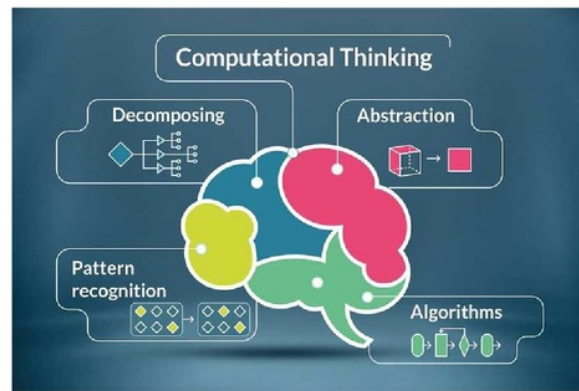


Figure 2. The elements of computational thinking.

Studies have been conducted to apply CT in schools, [1] has been done in K-12 [10] The study was to design standards of computational thinking curricula for K-12 education. The Delphi technique is used to collect different opinions and get agreement from thirteen expert discussions, including computer scientists, computer science educators, computer teachers and industry experts. The first draft of the Delphi questionnaire survey consisted of several themes: problem solving, problem decomposition, algorithm, data representation, data analysis, modelling and simulation, [1]straction, and automation and then developed CT competence indicator based on the research. After three rounds of surveys and round [1]table meetings, experts produced 49 essential competency indicators (13 for grades 1 through 6, 9 for grades 7 through 9 and 27 for grades 10 through 12) and 8 optional competency indicators (1 for grade 7 up to 9 and 7 for classes 10 to 12). The result is that the capabilities and training of CT differ in each level.

In the Computational Thinking Pedagogical Framework (CTPF) associated with the pedagogical framework includes four pedagogical experiences one of which is Tinkering (tweaking) [11] The experience of fiddling primarily separates and engages in changes and / or modifications to existing objects. These objects can be building blocks, puzzles, digital or electronic simulations, programming codes, and so on. During the tinkering, students do not build objects, digital or otherwise, but rather explore changes to existing objects and then consider the implications of change. These experiences may require students to use some of the basic concepts and skills learned, but new concepts and skills that [2]re being introduced can also be generated

In recent years, the availability of a free and user-friendly programming language has increased the interest of researchers and educators to explore how computational thinking can be introduced. Programming language is known as something hard to learn even many who believe only smart people who can operate it. However, the assumption is no longer appropriate, because now [9] has developed a relatively easy to learn programming language that is Scratch programming [12] Scratch is a new programming language released in May 2007. Scratch supports the development of computer games, interactive stories, graphics and computer animation and other multimedia projects. One of the programming languages that uses the Tinkering method is Scratch. Below is an example of an overview of the use of the Scratch programming language.

Scratch programming does not use text form but in graphical form to put the codes used. Graphic forms are made in such a way that they form like puzzle pieces known as code blocks. To create a

Scratch program is like composing a puzzle. Code blocks can only be arranged if they can be interconnected. Thus the program created is protected from syntax errors. In the Scratch there are three types of code blocks and each can be a relation. The code block provided by Scratch 100 is further inserted into the 8 categories of code blocks.

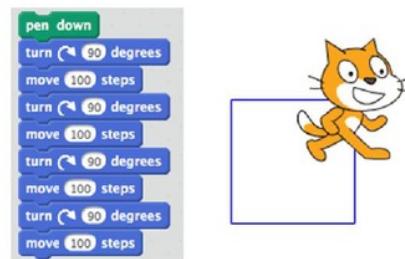


Figure 3. Draw a quadrilateral using Scratch.



Figure 4. Block the script used as the base of script writing.

To create a Scratch app, required an object known as a sprite. A Scratch app can consist of many sprites. Each Sprite has several scripts, costumes and sound types. Scripts are programs for creating applications. The costume is the image that will be displayed. Medium types of sounds are the sounds that will be played in the app. The program results will be shown in an area called stage / stage.

In storing Scratch data using variable code block. Just like any other programming variables in Scratch can be created as local variables or global variables. This variable code block can be used to control application execution. The value of the variable code block can be displayed on the stage / stage. If no longer needed, the variable block code can be deleted. As for numerical manipulation, Scratch has a block of numeric codes. In this block of code there are several blocks of code for mathematical operations, be it arithmetic or logic. To support the in-app process, Scratch provides the control blocks used for looping and conditional processes. It can also be used to control and coordinate scripting activities by sending and receiving messages between sprites.

Learning using the Scratch programming language will make it easier for students to understand the exact programming concepts and ensure that novice programmers learn the right way to structure and formulate programming logic. Because Scratch is not a text-based programming language it is not necessary to follow a complex set of syntax rules, because misconduct that does not match the syntax rules of the programming language will cause the application to not run. Scratch simplifies application

development using the same basic logic and programming concepts with other programming languages

To measure the extent of CT capability use measurement scale, developed for the purpose of determining the level of computational thinking ability (CTS) of students [13] The CTS used is a five-point Likert scale and consists of 29 items from 5 developed indicators. The validity and reliability of the scale has been studied by exploratory factor analysis, confirmatory factor analysis, goods specificity analysis, internal consistency coefficient and analysis of firmness. As a result of the analysis, it has been concluded that the measurement scale is valid and reliable as a tool that can measure students' computational thinking skills. However, the validity and reliability of such scales can only be used in certain groups. If it is to be used in different groups, renewals should be made in accordance with the group.

4. Conclusion

Computational Thinking (CT) is basically problem solving, designing systems, and understanding human behaviour, by drawing on fundamental concepts to computer science which can be done by humans or machines (computers). The goal of embedding computational thinking skills is to prepare students with the ability to produce computational solutions to problem solving. CT has strong principles and characteristics that serve as a basis for CT development and its implementation at several levels of education that have applied CT a lot in the curriculum.

Several levels of education have implemented a CT-based curriculum and have undertaken research that supports the further development of CT. Implementation of CT at the level of education is completed to facilitate students to solve problems using computers with programming language that is easy to understand and use. Then some research has been conducted to develop the scale of assessment to measure the level of ability of CT in students. The scale is considered valid for certain groups only.

Further CT development is CT can be the basis for the development of ICT curriculum nationally, because at this time ICT learning using different curriculum concepts of guidance. It thus must be made a curriculum design that is packed in such a way for the more structured ICT counselling process.

5. References

- [18] Wing, *Computational Thinking*, Communications of the ACM 49(3), pp. 33-35, 2006.
- [2] Sze Yee Lye, Joyce Hwee Ling Koh, *Review on teaching and learning of computational thinking through programming: What is next for K-12?* National Institute of Education, Nanyang Technological University, Singapore, 1 Nanyang Walk, Singapore 637616.
- [29] Singapore Computers in Human Behavior 41 (2014) 51-61.
- [3] James J. Lu, George H. L. Fletcher, *Thinking About Computational Thinking*, SIGCSE '09, March 3-7, 2009, Chattanooga, Tennessee, USA, Copyright 2009 ACM 978-1-60558-183-5/09/03.
- [4] Denning Peter J, *Great principles of computing*, Communications of the ACM-Blueprint for the future of high-performance networking, Volume 46 Issue 11, November 2003, Pages 15-20.
- [5] Perković Ljubomir, Seif Amber, *Computational thinking across the curriculum: A conceptual Framework*, 2009. <http://compthink.cs.depaul.edu/FinalFramework.pdf>, Accessed October 12, 2015.

- [6] Cuny Jan, Snyder Larry and Wing Jeannette M, *Demystifying Computational Thinking for Non-Computer Scientists*, work in progress, 2010.
- [7] Ana Liz Souto O. de Araujo, Wilkerson L. Andrade, Dalton D. Serey Gutierrez, *A Systematic Mapping Study on Assessing Computational Thinking Abilities*, Software Practices Laboratory Federal University of Campina Grande Department of Exact Sciences Federal University of Paraíba Paraíba, Brazil. Wing.
- [8] Osman Yasar, *The Essence of Computational Thinking*, State University of New York, Brockport.
- [9] Marcos Roman-Gonzalez, Juan-Carlos Perez-Gonzalez, Carmen Jimenez-Fernandez, *Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test*, Computers in Human Behavior xxx (2016) 1e1-12.
- [10] Hui-Chi, Chuang; Chiu-Fan, Hu; Cheng-Chih, Wu; Yu-Tzu, Lin, *Computational Thinking Curriculum for K-12 Education-A Delphi Survey*, 2015 International Conference on Learning and Teaching in Computing and Engineering.
- [11] Donna Kotsopoulos & Lisa Floyd & Steven Khan & Immaculate Kizito Namukasa & Sowmya Somanath & Jessica Weber & Chris Yiu, *A Pedagogical Framework for Computational Thinking*.
- [12] Ekohariadi, 2015, Ibnu Febry Kurniawan, Ricky Eka Putra, Pemrograman Visual Menggunakan Scratch, Unesa University Press
- [13] Özgen KORKMAZ TURKEY Recep ÇAKIR, *A validity and reliability study of the Computational Thinking Scales (CTS)*.

The concept of computational thinking toward information and communication technology learning

ORIGINALITY REPORT

23%

SIMILARITY INDEX

PRIMARY SOURCES

- 1

Hui-Chi Chuang, Chiu-Fan Hu, Cheng-Chih Wu, Yu-Tzu Lin. "Computational Thinking Curriculum for K-12 Education -- A Delphi Survey", 2015 International Conference on Learning and Teaching in Computing and Engineering, 2015

78 words — 2%

Crossref
- 2

Lye, Sze Yee, and Joyce Hwee Ling Koh. "Review on teaching and learning of computational thinking through programming: What is next for K-12?", Computers in Human Behavior, 2014.

65 words — 2%

Crossref
- 3

Qurrotul Anfa, Fida Rachmadiarti, Winarsih, Tien Aminatun. " Empirical Validity of Collaborative-Learning Student Worksheet on Ecology Material to Practice Student's Scientific Literacy Skill of Ten Grade Senior High School ", Journal of Physics: Conference Series, 2019

59 words — 2%

Crossref
- 4

Osman Yasar. "The Essence of Computational Thinking", Computing in Science & Engineering, 2017

33 words — 1%

Crossref
- 5

Christian P. Brackmann, Marcos Román-González, Gregorio Robles, Jesús Moreno-León, Ana Casali, Dante Barone. "Development of Computational Thinking Skills through Unplugged Activities in Primary School", Proceedings of the 12th Workshop on Primary and Secondary Computing Education - WiPSCE '17, 2017

31 words — 1%

Crossref
- 6

shura.shu.ac.uk

27 words — 1%

- 7 Alberto Ferrari, Agostino Poggi, Michele Tomaiuolo. "chapter 18 Computational Thinking in Innovative Computational Environments and Coding", IGI Global, 2019

Crossref

27 words — 1%

- 8 Martin Mohapl, Jana Prazakova, Martin Hejl. "Impact on Total Costs and Construction Period of Residential Buildings in Prague", IOP Conference Series: Earth and Environmental Science, 2019

Crossref

27 words — 1%

- 9 book.pdfchm.net

Internet

26 words — 1%

- 10 Ana Liz Souto O. de Araujo, Wilkerson L. Andrade, Dalton D. Serey Guerrero. "A systematic mapping study on assessing computational thinking abilities", 2016 IEEE Frontiers in Education Conference (FIE), 2016

Crossref

25 words — 1%

- 11 Ana Liz Souto O. Araujo, Wilkerson L. Andrade, Dalton D. Serey Guerrero, Monilly Ramos Araujo Melo. "How Many Abilities Can We Measure in Computational Thinking?", Proceedings of the 50th ACM Technical Symposium on Computer Science Education - SIGCSE '19, 2019

Crossref

22 words — 1%

- 12 Leovy Echeverria, Ruth Cobos, Mario Morales, Fernando Moreno, Victor Negrete. "Promoting Computational Thinking Skills in Primary School Students to Improve Learning of Geometry", 2019 IEEE Global Engineering Education Conference (EDUCON), 2019

Crossref

21 words — 1%

- 13 Peter J. Denning. "Great principles of computing", Communications of the ACM, 2003

Crossref

21 words — 1%

14	pages.cpsc.ucalgary.ca Internet	20 words — 1%
15	kanemune.eplang.jp Internet	19 words — 1%
16	Arzu Deveci Topal, Esra Çoban Budak, Aynur Kolburan Geçer. "The effect of algorithm teaching on the problem-solving skills of deaf-hard hearing students", Program, 2017 Crossref	19 words — 1%
17	Danli Wang, Tingting Wang, Zhen Liu. "A Tangible Programming Tool for Children to Cultivate Computational Thinking", The Scientific World Journal, 2014 Crossref	18 words — 1%
18	www.jitae.org Internet	15 words — < 1%
19	kimpearson.net Internet	15 words — < 1%
20	danielrichardsdotnet.files.wordpress.com Internet	14 words — < 1%
21	sigite2012.sigite.org Internet	13 words — < 1%
22	Ana M ^a Pinto-Llorente, Sonia Casillas-Martín, Marcos Cabezas-González, Francisco José García-Peñalvo. "Building, coding and programming 3D models via a visual programming environment", Quality & Quantity, 2017 Crossref	13 words — < 1%
23	learntechlib.org Internet	13 words — < 1%
24	Aman Yadav, Chris Mayfield, Ninger Zhou, Susanne Hambrusch, John T. Korb. "Computational	13 words — < 1%

-
- 25 doras.dcu.ie 12 words — < 1%
Internet
-
- 26 Smart Innovation Systems and Technologies, 2015. 12 words — < 1%
Crossref
-
- 27 Valerie J. Shute, Chen Sun, Jodi Asbell-Clarke.
"Demystifying computational thinking", Educational
Research Review, 2017 11 words — < 1%
Crossref
-
- 28 mycomputer.computer.org 11 words — < 1%
Internet
-
- 29 portal.acm.org 11 words — < 1%
Internet
-
- 30 Klinge Orlando Villalba Condori. "chapter 4
Teaching Formation to Develop Computational
Thinking", IGI Global, 2018 10 words — < 1%
Crossref
-
- 31 Marcos Román-González, Juan-Carlos Pérez-
González, Jesús Moreno-León, Gregorio Robles.
"Extending the nomological network of computational thinking with
non-cognitive factors", Computers in Human Behavior, 2018 9 words — < 1%
Crossref
-
- 32 www.terecop.eu 8 words — < 1%
Internet
-
- 33 Takam Djambong, Viktor Freiman, Simon Gauvin,
Martine Paquet, Mario Chiasson. "Chapter 12
Measurement of Computational Thinking in K-12 Education: The
Need for Innovative Practices", Springer Nature, 2018 8 words — < 1%
Crossref
-
- 34 James J. Lu. "Thinking about computational thinking", 8 words — < 1%

35

Woonhee Sung, Junghyun Ahn, John B. Black.
"Introducing Computational Thinking to Young Learners: Practicing Computational Perspectives Through Embodiment in Mathematics Education", Technology, Knowledge and Learning, 2017

6 words — < 1%

Crossref