

Systematizing ICT Education Curriculum for Developing Computational Thinking: Case Studies of Curricula in the United States, Australia, and the United Kingdom

Satoshi Kusaka¹

¹Graduate School of Education, Naruto University of Education, Japan

Correspondence: Satoshi Kusaka, Graduate School of Education, Naruto University of Education, Japan. E-mail: kusaka@ksm.world

Received: July 9, 2021

Accepted: August 17, 2021

Online Published: August 30, 2021

doi:10.5539/jel.v10n5p76

URL: <https://doi.org/10.5539/jel.v10n5p76>

Abstract

Information and Communication Technology (ICT) education has been actively implemented around the world recently. ICT curriculum in schools is intended to improve students' programming-oriented thinking rather than to train them as programmers. The core of ICT education is 'computational thinking'. Computational thinking is taking an approach to solving problems, designing systems and understanding human behaviour that draws on concepts. Various countries try to develop ICT curriculum based on this concept. However, unlike traditional subjects, the ICT curriculum has yet to create a unified structure. This research sheds light on the characteristics of ICT education curricula by comparing and contrasting curricula in the United Kingdom (UK), the United States and Australia. Finally, a common framework is proposed on the basis of the discussion. Furthermore, while ICT education is often addressed solely from the standpoint of science and technology, a social science perspective is needed. Therefore, the importance of social science in ICT education will also be clarified.

Keywords: computational thinking, ICT education, curriculum of ICT education

1. Introduction

Information and Communication Technology (ICT) education is gathering momentum at both the elementary and secondary levels worldwide. In 2014, the United Kingdom adopted a new ICT education system named 'Computing', which focuses on programming education. Additionally, ICT education focussing on programming has been officially introduced into many countries such as the United States, Australia and Japan. The aim of programming education during compulsory education, on the other hand, is not to train programmers. The goal of programming education in elementary schools is not to train programmers, but to improve programming-oriented thinking in children, just as music as a subject is not intended to train pianists or physical education to train professional athletes. The ability to think logically is described as this form of thought. This involves thinking about things like what actions are needed to complete a mission, how to combine symbols that correlate with each of these actions, and whether rearranging these symbols will get you closer to your intended aim of realising the chain of actions you are working on. Furthermore, programming-oriented thinking is focused on computational thinking, a concept that acknowledges the existence of a relationship between programming and logical thinking while also defining what that relationship implies. Essentially, programming-oriented thinking, which is a subset of computational thinking, seeks to develop the above term and provides a programming-related curriculum in ICT education. In 2006, Jeannette Wing described computational thinking as 'solving problems, designing systems and understanding human behaviour by drawing on fundamental computer science principles' (Wing, 2006, p. 33), and it has since gained widespread acceptance. Computational thinking, on the other hand, is often thought of as a way of thinking that is required only of natural and computer scientists; however, this is not the case. It will also become increasingly relevant in the social sciences. As the use of computers and Artificial Intelligence (AI) has become more widespread in our lives, it is expected that analytical thinking will become increasingly important in all fields. Additionally, beginning in 2021, the category of computational thinking will be added to mathematics in the Programme for International Student Assessment (PISA).

However, research and implementation in schools in this field have only recently begun, and all countries are

looking for a suitable ICT curriculum focused on the principle of computational thinking. Until now, an integrated framework has not existed as is seen in other existing subjects.

It has been said that many existing occupations will disappear after 2045, a time where it is predicted that AI will exceed human capabilities. Determining children's abilities in information technology education and systemising this into a curriculum is a pressing issue.

2. Research Objectives

This research sheds light on the characteristics of ICT education curricula by comparing and contrasting curricula in the United Kingdom (UK), the United States and Australia. Finally, a common curriculum framework is proposed on the basis of the discussion.

3. Methodology

The following steps will be taken to achieve the research objectives.

- (1) To deliberate upon and examine the theory of computational thinking based on previous research.
- (2) To conduct a comparative analysis of the concept of Computational Thinking in the UK, the United States, and Australia.
- (3) To collate a curriculum related to information technology education in these three countries.
- (4) Based on steps (1), (2) and (3), to examine the characteristics of computational thinking and ICT education curricula and to propose a common framework for the curriculum by focusing on the similarities and differences between the three countries.

3.1 Reason for Choosing the Three Countries

The reason why these three countries are chosen is that the UK and Australia implement ICT education as a school subject from the first year of primary education through a nationwide curriculum. In the U.S., there is no nationally standardised curriculum. However, the Computer Science Teachers Association (CSTA) has developed a curriculum for information education, the CSTA K-12 Computer Science Standards (CSTA, 2011), which leads programming education in the United States and worldwide. Therefore, the United States was included as a target country.

4. Computational Thinking

4.1 Definition of Computational Thinking

Regarding the term 'computational thinking', Computing at School (CAS) (2015a) stated that it was first used by Papert and subsequently popularised by Wing in 2006. Since then, many researchers have attempted to define computational thinking. The main definitions of this concept have been summarised in Table 1. Wing (2006, p. 33) defined it as follows: Computational thinking involves solving problems, designing systems and understanding human behaviour by drawing on the concepts fundamental to computer science. It is a fundamental skill for everyone, not just for computer scientists. To reading, writing and arithmetic, we should add computational thinking to every child's analytical ability.

Furthermore, Wing (2006, p. 34) defines computational thinking as 'thinking like a computer scientist'. It is more than being able to programme a computer. It necessitates abstraction at multiple levels. In other words, this equates to a way of thinking for humans that require multiple levels of abstract thinking, something computers cannot do. Computational thinking is a method used by humans in problem-solving. It is not aimed at making humans think like computers.

Moreover, Wing (2008) stated that the essence of computational thinking is an abstraction and further added that abstraction is a spiritual tool of computing, and physical tools can amplify the power of mental tools. Computing is the automation of abstract human thinking. Wing indicated that the two stages of abstraction and automation (computerisation) in computational thinking are its most basic components. Likewise, Yadav et al. (2014) also defined computational thinking as a mental process for abstracting problems and producing automated solutions.

Definitions tend to be focused on the perception of objects and the process of problem-solving as a whole. In other words, it can be concluded that the computational thinking-based learning aims at supporting the educational and learning processes of its targeted learners. However, Wing viewed computational thinking differently and stated that there is no common perception of the concept itself as there is no clear definition of it. Research and discussion concerning its definition are ongoing (Hu, 2011). Additionally, many researchers pointed out that it's unnecessary to define computational thinking unequivocally: instead, establishing a relationship within the corpus with this concept should be aimed (Voogt et al., 2015). Through this establishment

of relationships within the corpus, the concept of computational thinking can be consequently systematised, and the content to be handled within education practised in schools and the arrangement of learning contents in each year level will be established.

Definitions tend to be focused on the perception of objects and the process of problem-solving as a whole. In other words, it can be concluded that the computational thinking-based learning aims at supporting the educational and learning processes of its targeted learners. However, Wing viewed computational thinking differently and stated that there is no common perception of the concept itself as there is no clear definition of it. Research and discussion concerning its definition are ongoing (Hu, 2011). Additionally, many researchers pointed out that it's unnecessary to define computational thinking unequivocally: instead, establishing a relationship within the corpus with this concept should be aimed (Voogt et al., 2015). Through this establishment of relationships within the corpus, the concept of computational thinking can be consequently systematised, and the content to be handled within education practised in schools and the arrangement of learning contents in each year level will be established.

Table 1. Contrasting definitions of computational thinking

Wing (2006, 2008)	Computational thinking involves solving problems, designing systems and understanding human behaviour that draws on concepts fundamental to computing.
Denning (2009)	Computational thinking includes thinking with many levels of abstractions, using mathematics to develop algorithms and examining how well solution scales across different sizes of problems.
Furber (2012)	Computational thinking is the process of recognising aspects of computation in the world that surround us and applying tools and techniques from computer science to understand and reason for both natural and artificial systems and processes.
Wing (2014)	Computational thinking is the thought process used to develop a problem and express its solution or solutions in terms a computer can apply effectively.
Yadav et al. (2014)	The mental process for the abstracting problems and the creation of automatable solutions.

4.2 Importance of Computational Thinking in Social Science

In the social sciences, computational thinking is needed for various reasons. To begin with, social science is a way of thinking that helps one to objectively solve problems. Computational thinking, according to Park and Green (2019), is a problem-solving skill that becomes more advanced as one grows older. The people who learn computational thinking will get abilities such as communication and collaboration skills, motivation, complex problem-solving skills, abstraction and transfer (Rich & Hodges, 2017). In other words, computational thinking is not a special realm of computer scientists, and not everyone should think like a computer scientist. There may have been an image of 'people who know programming is an excellent engineer' until recent years. However, we are now in an era where people who know programming and programme-oriented thinking can be seen as people who can do high-quality work in general. Therefore, computational thinking is a fundamental resource for society as a whole. It will be useful both in their respective fields and in the larger public society.

Second, social science is the study of seeking truth in society. Just as the natural sciences include physics, chemistry and biology, the social sciences include political science, economics, education, sociology, etc. For example, social problems, such as why wars happen, why some countries are rich and others are poor, are studied in social sciences. Google For Education defined computational thinking as a 'problem-solving process that includes a numerous characteristics, such as logically ordering and analysing data and creating solutions using a series of algorithms, and dispositions, such as the ability to deal with complexity and open-ended problems confidently'.

Lastly, Security policy in ICT education cannot be discussed without the context of social sciences, like domestic and international politics. Therefore, it is necessary to discuss factors other than technological science. When deciding the security policy, it is necessary to carefully discuss how economics, management and law, including politics, should deal with security. As a result, it is essential to have the concept of computational thinking in the social sciences.

4.3 The Concept of Computational Thinking in the Three Countries

Depending on the country, computational thinking is interpreted differently. As a result, this study would compare and contrast the idea of computational thinking in the three countries studied. In the UK, the National Curriculum for Computing, a new subject replacing standard ICT, was announced in 2013 and has been implemented since 2014. At the beginning of the National Curriculum (DfE, 2013), it was described as 'A

high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world'. Additionally, CAS, a community of teachers, academics and those connected to the industry who supports computer education within British schools, has delineated computational thinking (CAS, 2015b). Computational thinking is defined as a collection of several intellectual abilities that convert complex, cluttered and only partly visible realisation problems with instructions that can be tackled solely by an intelligent computer (BCS, 2014). Its ability can be applied not only to problem-solving using computers but also to problem-solving that involves humans. CAS considers this collection of abilities as logical reasoning, in which facts are analysed, predictions are confirmed by thought and results are visualised. Moreover, this is classified into five abilities: 'Abstraction', 'Decomposition', 'Algorithmic Thinking', 'Evaluation' and 'Generalisation'.

In Australia, computational thinking is explained as follows: computational thinking is a problem-solving method applied to create solutions that can be implemented using digital technologies. It involves integrating strategies, such as organising data logically, breaking down problems in parts, interpreting patterns and models and designing and implementing algorithms (Australian Curriculum, Assessment and Reporting Authority (ACARA, 2013).

Since 2016, the topic of 'technology' has been gradually introduced in each state and territory. This course prepares students for a better future by teaching them about processes, architecture and computational thinking, as well as project management. Under this context, the principles of 'Abstraction, Data Collection, Representation and Interpretation', 'Specification, Algorithms and Implementation' are held to correlate with computational thinking. The CSTA in the United States has established the CSTA K-12 Computer Science Standards and curriculum for information technology education (CSTA, 2012). This standard defines computer science as 'research on computers and algorithmic processes including its principle, hardware and software design, its impact on society and places it at the core. Learning contents are divided into five areas: Computational Thinking, Collaboration, Computing Practise and Programming, Computers and Communications Devices and Community, Global and Ethical Impacts. Computational thinking is subdivided into 'Problem Solving', 'Algorithms', 'Data Representation', 'Models and Simulations', 'Abstraction' and 'Utilisation Other Subjects'. The curriculum was revised in 2017, and the term computational thinking is no longer used, but the idea remains.

Table 2. Components of computational thinking

UK (Computer Progress Pathway)	Australian Curriculum	USA (CASTA)
Abstraction	Abstraction	Abstraction
Algorithmic Thinking Decomposition	Specification, Algorithms and implementation	Algorithms
	Data collection, Representation and Interpretation	Data Representation
Generalization		Models and Simulations
		Utilization in Other Subjects
		Problem Solving
Evaluation		

4.4 Definition of Computational Thinking in PISA

Wing (2010) stated that computational thinking is the new literacy of the 21st century. PISA's 2021 mathematics test will add questions to measure competencies such as logical thinking and problem-solving skills, which pertain to computational thinking. As for why this new question was added, PISA explained that computational thought is the foundation of computer science, which encompasses a several topics such as data science, cryptography, networking, machine learning and robotics. Additionally, the PISA 2021 mathematics framework points to four areas in which can provide rich data. Computational thinking is selected as one of the areas. PISA's 2021 specification framework mentions as follows:

Pattern recognition, designing and using an abstraction, pattern decomposition, deciding which (if any) computing methods may be used in analysing or solving a problem and identifying algorithms as part of a detailed solution are all computational thinking skills. Pattern recognition, designing and using an abstraction, pattern decomposition, deciding which (if any) computing methods may be used in analysing or solving a problem and identifying algorithms as part of a detailed solution are all computational thinking skills. As part of their problem-solving practise, the students should also possess and show computational thinking skills.

The concrete components of computational thinking cited by PISA are 'Abstraction', 'Algorithmic Thinking', 'Automation', 'Decomposition' and 'Generalisation'. The 'Abstraction' and 'Algorithms' contained in all three

countries that were compared earlier are also included in PISA's components of computational thinking. Additionally, 'Automation' is one of the basic elements of computational thinking, according to Wing. 'Decomposition' is a component that only Australia includes as a component of computational thinking. It is possible to interpret it as Utilisation of 'Abstraction', one of the fundamentals of computational thinking or as a means of facilitating 'Abstraction'. Moreover, PISA emphasises the importance of computational thinking in the existing subject of 'Mathematics'. However, additional research will be necessary on how learning contents within mathematics interact with computational thinking and how mathematical inference complements computational thinking (Gadanidis, 2015; Rambally, 2017).

Up to now, general concepts of computational thinking have been discussed. However, further considerations may be difficult as to how ICT curricula realise computational thinking within actual school settings and teach computational thinking to children in actual classrooms. In the next section, curricula regarding information technology education in the three countries will be compared and elucidated.

5. Three Countries' Curricula Regarding ICT Education

The curricula regarding ICT education in the three countries have the idea of computational thinking at their core and have established learning contents developing abilities such as abstraction, problem analysis, algorithms, data utilisation, evaluation and collaboration. First, the composition areas shown in the curricula regarding ICT education in the three abovementioned countries will be compared and analysed.

5.1 UK

QuickStart Computing, which was created by CAS and two types for primary and secondary education, is an actual teaching guide (CAS, 2014) for all teachers. Furthermore, a learning content system that provides the national curriculum with more detail is outlined as Computing Progression Pathways (CAS, 2015a). This Computing Progression Pathways defines the 153 pieces of content to be learnt from Grade 1 to Grade 11 and divides similar content into six different categories: 1) Algorithms 2) Programming and Development 3) Data and Data Representation, 4) Hardware and Processing, 5) Communication and Networks and 6) Information Technology.

5.2 Australia

In Australia, the ACARA have developed the National Curriculum. The abilities in Systems Thinking, Design Thinking, Computational Thinking and Project Management are placed at the forefront of information technology education to create a better future. Existing subjects include Design and Technologies and Digital Technologies. The latter subject is on learning contents related to information technology education and consists of five areas: 1) Abstraction 2) Data Collection, Representation and Interpretation 3) Specification, Algorithms and Implementation, 4) Digital Systems and 5) Interactions and Effects.

5.3 United States

The International Society for Technology in Education (ISTA) and the Computer Science Teachers Association (CASTA) in the United States have been tackling organising the concept and capabilities of computational thinking, and the results were published under the name of ISTE & CSTA (Barr & Stephenson, 2011). In 2017, the revised CSTA was composed of five areas: 1) Computing Systems 2) Networks and the Internet 3) Data and Analysis 4) Algorithms and Programming and 5) Impacts of Computing.

Table 3. Components of ICT education in three countries

U.K.	Australia	USA
1. Algorithm	1. Abstraction	1. Computing Systems
2. Programming & Development	2. Data Collection, Representation and Interpretation	2. Networks and the Internet
3. Data & Data Representation	3. Specification, Algorithms and Implementation,	3. Data and Analysis
4. Hardware & Processing	4. Digital Systems and	4. Algorithms and Programming
5. Communication & Networks	5. Interactions and Effects.	5. Impacts of Computing
6. Information Technology		

Table 3 summarises the areas within the information technology education in these three countries. All three countries have three elements in common: 'Algorithms', 'Programming' and 'Data'. The 'Networks' was cited in only the United States and the UK. The areas exclusively established in the UK are 'Hardware & Processing' and 'Information Technology'. Similarly, 'Digital Systems' and 'Interactions and Effects' are specific only to

Australia, and 'Computing Systems' as well as 'Impacts of Computing' are specific only to the United States. However, even if the names of the areas are different, there is a possibility that the contents that are handled may be similar. In the next section, this will be clarified through a detailed analysis of the learning contents of each area. Ultimately, newly integrated learning areas in information technology education and learning elements divided into each area will be proposed.

6. Proposition of ICT Education Learning Domain and Content

Since the setting of the learning domain is different in the three countries, it is not easy to directly compare the learning contents and systematisation established in the areas. Therefore, the classification was conducted by analysing the following sets of detailed learning contents written in the curricula of the three countries: Pathways (consisting of 153 items), AC (44 items) and Standards (167 items). because of this classification, the following seven domains could be defined: 'A. Computational Thinking', 'B. Algorithms', 'C. Programmes' 'D. Data Utilisation', 'E. Networks', 'F. Evaluation' and 'G. Information Handling'. The elements classified into each area will be explained by focussing on the study content and the learning year level listed in the curriculum of each country.

The main purpose of ICT education is to foster computational thinking. Therefore, computational thinking is brought as the first. The element of 'Abstraction', 'Decomposition', 'Generalisation' and 'Interaction' were collected and made into a single domain: 'A. Computational Thinking'. These elements are similar to the computational thinking framework contents designated by the three countries and shown in Table 2. 'Abstraction' and 'Decomposition' are introduced in the lower grades in the UK, and in the higher grades in Australia and the United States. 'Generalisation' is introduced in the lower grades in all three countries. The interaction is concerned with the working range of computers and humans, and their interaction. Because it has been taken up in the UK and the United States, it is decided to include 'computational thinking'.

'B. Algorithms' is a component that has been established as a domain in all three countries. Problem-solving contents corresponding to age using both patterns and beginner programmes are included in the curricula of all three countries. Moreover, representing algorithms with structured sentences and understanding that there are different solution methods to the same problem are included in all countries' curricula. Based on the above, in the domain of the Algorithms, three elements, 'Algorithms and Programmes', 'Algorithms Generation and Expression' and 'Multiple Solution Methods', are cited as components belonging to 'B. Algorithms'.

With regard to 'C. Programming', the only UK and the United States have established 'Programming' as one domain, but its learning content is included similarly in the Australian curriculum. Since Programming learning content is included in the curricula of the three countries, 'Programming' has been established as a domain, and as such, includes five elements from all three countries that have been incorporated. As for 'Visual Language', all three countries have introduced programme creation activities with visual languages early on. As for 'Text Language', learning to create programmes using general-purpose programme languages is included from the primary level in the UK and the middle grades in Australia and the United States. 'Sequence', 'Iteration' and 'Selection' as they pertain to the differences in iterative processing are included in both the primary and middle grades in the UK, the primary levels in Australia, and the middle grades in the United States, respectively. Concerning 'Variables and Data Types', learning to set variables and select appropriate data types is included in both the primary and middle grades in the UK and Australia and in the middle grades in the United States.

In the case of 'D. Data Utilisation', it was characteristic in all three countries to learn about the binary system, which relates to how to express information with two variables (0 and 1), from the lower grades. Also, since the expression of the hexadecimal and binary systems is also handled in all three countries, it was made into one element related to Data Utilisation.

Concerning 'E. Evaluation', learning contents are diversely depending on the country, but it includes the contents to assess future risks in existing information systems, prepare tests for that purpose and improve problem-solving methods based on the feedback provided. As for various learning contents related to Evaluation, 'Evaluation Criteria' element was established. Additionally, debugging, which is a process for finding and fixing bugs and defects in computer programmes and electrical equipment and having the system operate as specified, was adopted as one of the elements.

Concerning 'F. Networks', as learning contents related to 'Network Structure and Functioning' is included through all year levels in all the three countries, it was also established as one of the elements.

As mentioned earlier, ICT education should not be taught only from technological science but should also include a social science perspective. 'G. Information Handling' is directly related to this point. Each country

starts information ethics and information safety education from the lower elementary school grades in this domain. In particular, both the UK and the United States emphasise precautions as ‘Users Responsibility’ for using a computer, especially an internet. Additionally, when it comes to intellectual property rights, issues of software licencing such as the difference between commercial and open-source software have been taken into learning content, which is characteristic of the United States curriculum. Also, the digital divide is treated as a regional problem and as racial and gender equality. As mentioned earlier, ‘Informational Ethics’, ‘Informational Society’, ‘Informational Reliability’ and ‘Information Security’, which are contents handled throughout the school year in this area, have also been established as elements.

Table 3. Proposition of domains and components of ICT education

Domains	Component
A. Computational thinking	Abstraction, Decomposition, Generalization, Interaction
B. Algorithm	Algorithm and program, Creation and representation of algorithm, Multiple process of solution
C. Programming	Visual programming languages, text-based programing languages, Sequence, iteration and selection, Variables and data type
D. Data utilization	Data retrieval, Data collection, management, and analysis, Databases, Binaries, Numerical representation
E. Evaluation	Evaluation criteria, Debug
F. Network	Structure and functions of network
G. Information Handling	Informational ethics, Informational society, Informational reliability, Information security

7. Conclusion

As we examine the world’s primary education upon the advent of the fully-fledged digital age, various discussions have been held about the new abilities that should be acquired henceforth, and thus many initiatives have been put into practise. The key ability is computational thinking. Although ICT education is the most necessary subject for fostering computational thinking, the learning contents are still in the research process, and unlike other conventional subjects, there has not been a globally common curriculum. Children do not suddenly acquire computational thinking. They must learn in a step-by-step manner. For this reason, a systematic curriculum is indispensable.

We attempted to develop a shared structure in this study by comparing and contrasting the information technology education curricula in three countries: the United States, Australia and the UK. As a result of the analysis, we could reconstruct the curriculum into seven domains: ‘A. Computational Thinking’, ‘B. Algorithms’, ‘C. Programmes’, ‘D. Data Utilisation’, ‘E. Networks’, ‘F. Evaluation’ and ‘G. Information Handling’. Additionally, we extracted contents that are to be learnt under each domain as components.

With the rapid progress of the digital society recently, each country is trying to develop information technology education curricula. However, there are still not many countries that have officially devised concrete curricula or lesson content, as this is something entrusted to the discretion of each school in question. This study proposed a framework for information technology education by comparing the curricula in the three countries where ICT education is advanced and analysing their similarities and differences. Additionally, the importance of ICT education from a social science perspective was also clarified. This research has been limited to the selection of domains and the components included in these domains. In the future, however, in addition to conducting the basic study on curricula as seen in this study, by conducting research that combines the implementation of various curricula, it is expected that learning contents will be systematically arranged according to each grade, leading to the systematisation of ICT education curriculum.

References

- ACARA. (2013). *Digital Technologies Foundation to Year 10 scope and sequence*. Retrieved May 15, 2021, from <http://www.acara.edu.au/technologies.html>
- 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. <https://doi.org/10.1145/1929887.1929905>
- BCS. (2014). *The Chartered Institute for IT (2014) Call for evidence - UK Digital Skills Taskforce*. Retrieved May 20, 2021, from <http://policy.bcs.org/sites/policy.bcs.org/files/BCS%20response%20to%20UKDST%20call%20for%20evidence%20final.pdf>

- CAS. (2014). *QuickStart Computing*. Retrieved May 29, 2016, from <http://www.quickstartcomputing.org>
- CAS. (2015a). *Computational Thinking—A Guide for teachers*. Retrieved May 15, 2021, from <http://community.computingatschool.org.uk/resources/2324>
- CAS. (2015b). *CAS Computing Progression Pathways KS1 (Y1) to KS3 (Y9) by topic*. Retrieved May 24, 2021, from <http://community.computingatschool.org.uk/resources/1692>
- CSTA. (2011). *CSTA (Computer Science Teachers Association) Standards Task Force: CSTA K-12 Computer Science Standards*. Retrieved May 15, 2021, from <https://www.csteachers.org/page/standards>
- CSTA. (2012). *CSTA K-12 Computer Science Standards: Mapped to Partnership for the 21st Century Essential Skills*. Retrieved May 10, 2021, from <https://csta.acm.org/Curriculum/sub/CurrFiles/CS>
- Denning, P. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, 52(6), 28–30. <https://doi.org/10.1145/1516046.1516054>
- DfE. (2013). *National curriculum in England: Computing programmes of study*. Retrieved May 15, 2021, from <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmesof-study>
- Furber, S. (2012). *Shut down or restart? The way forward for computing in UK schools*. Technical report, The Royal Society, London.
- Gadanidis, G. (2015). Coding as a Trojan Horse for Mathematics Education Reform. *Journal of Computers in Mathematics and Science Teaching*, 34(2), 155–173.
- Hu, C. (2011). *Computational thinking: What it might mean and what we might do about it* (pp. 223–227). Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education. <https://doi.org/10.1145/1999747.1999811>
- Park, Y., & Green, J. (2019). Review Bringing Computational Thinking into Science Education in the 21 Century. *Journal of Korean Earth Science Society*, 40(4), 340–352. <https://doi.org/10.5467/JKESS.2019.40.4.340>
- Rambally, G. (2017). Integrating Computational Thinking in Discrete Structures. In P. Rich & C. Hodges (Eds.), *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 99–119). Springer. https://doi.org/10.1007/978-3-319-52691-1_7
- Rich, P. J., & Hodges, C. B. (2017). *Emerging research, practice, and policy on computational thinking*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-52691-1>
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. <https://doi.org/10.1007/s10639-015-9412-6>
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49, 33–35. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Five deep questions in computing. *Communications of the ACM*, 51(1), 58–60. <https://doi.org/10.1145/1327452.1327479>
- Wing, J. M. (2010). *Research Notebook: Computational thinking: What and Why?* Link Magazine. Retrieved May 5, 2021, from <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why/>
- Wing, J. M. (2014). *Computational thinking benefits society*. 40th Anniversary Blog of Social Issues in Computing. Retrieved May 15, 2021, from <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>
- 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), 1–16. <https://doi.org/10.1145/2576872>

Copyrights

Copyright for this article is retained by the author, with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).