7

PATHWAYS OF COMPUTING EDUCATION: FORMAL AND INFORMAL APPROACHES

Chee-Kit Looi, Shiau-Wei Chan, Peter Seow, Longkai Wu, and Bimlesh Wadhwa

INTRODUCTION

Today, computing is becoming gradually more essential to our society. In many countries, it has been introduced into compulsory schooling (K–12) education, including in the form of STEM (Science, Technology, Engineering, and Mathematics) education. Manches and Plowman (2017) argue that computing education should be introduced to children from an early age. Computing is often taught in schools in the form of computational thinking (CT), which is about expressing solutions as algorithms or computational steps that can be executed using a computer (CSTA 2016). CT is advocated as a universal competence that can prepare children for future challenges in a growing digital world (Voogt et al. 2015). It is defined by Wing (2006) as "solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science" (33). In computing education, the definition of CT has two aims—(1) learning transferable knowledge from computing that can be utilized in everyday life and (2) employing computing concepts to promote computing work in other subjects (Guzdial 2015). The ideas of programming and algorithms that are mostly used to assess CT in K–12 computing education include variables, modularity, control, and algorithms (Alves, Von Wangenheim, and Hauck 2019).

Different learning pathways could meet students' needs in more effective ways than a single learning channel. Computationally talented learners show different abilities when learning coding or programming compared to regular learners; for instance, they can accelerate from block-based programming to text-based programming (Roman-Gonzalez et al. 2018). Learners with disabilities in computing require additional CT-specific supports to engage in the CT instruction, such as daily behavior reports, voluntary breaks, and after-school make-up time (Snodgrass, Israel, and Reese 2016).

From a macro perspective, formal and informal learning systems can be regarded as the two main learning pathways in computing education. Generally, a formal learning system can deliver a structure for systematic thinking and methods, while an informal learning system can assist students to increase their motivation and to recognize their interests (Bocconi et al. 2016). Many countries have infused computer science (CS) into K–12 formal education, including England, Estonia, Finland, Poland, and Slovakia. In addition to these formal learning environments, many informal offerings aim to increase children's knowledge and interest of CS. Even if they are not part of the curriculum, they also provide opportunities to communicate with CS as they mainly focus on coding or programming, for instance, the website Code.org and the programming clubs of Coder Dojo and Code Club. The learners can participate in these extracurricular activities that are not available in a regular classroom environment (Lunenburg 2010).

Computational tools, such as mobile devices, social media, programming, and robotic tangibles, provide various advantages for formal and informal learning (e.g., Dabbagh and Kitsantas 2012; Pereira, Baranauskas, and da Silva 2013; Khaddage, Müller, and Flintoff 2016; Burleson et al. 2018; Kjällander et al. 2018). It alters the way people learn as it allows novel and more instant methods in retrieving and generating knowledge through social interaction, new approaches of representation, and improved capability to cross space and time (Erstad and Sefton-Green 2013).

Guzdial (2015) had forecasted that a majority of students are likely to accomplish universal computing literacy through formal computing education pathways, namely elementary schools, secondary schools, and universities. Informal computing education (e.g., summer camps, online

programs, MOOCs, museums, coding boot camps, and after school programs) is not appropriate for every student. Earlier studies from Fields, Giang, and Kafai (2014) and Ho et al. (2015) concluded that informal computing learning was biased toward more wealthy and male students compared to formal learning. Nevertheless, informal computing education still plays a crucial role in affecting identity and computing perception among the students. In another research study by Guzdial et al. (2014), it was found that informal learning in the summer camps has a significant effect on the pipeline of formal education. Studies by Ehsan et al. (2018) and Hynes et al. (2019) have proposed that students' CT can be assessed through formal and informal learning settings.

LEARNING PATHWAYS OF COMPUTING EDUCATION

Differentiated instruction is described as "an organization of pedagogical practices that promotes reporting learners' needs from a learning situation" (Tahiri, Bennani, and Khalidi Idrissi 2017, 200). It presents several advantages to the learners, such as giving learning content modified to the preferences of learners, including all the capabilities desired, showing the learner the most suitable learning path, and making learning activities and situations in a meaningful way. In other words, this instruction is applied based on examining the learners' diversity, considering them, and augmenting the learning conditions. Differentiated instruction is required in the classroom as the features of learners are heterogeneous and diverse in terms of education, skills, learning styles, needs, sociocultural backgrounds, learning profiles, level of engagement, and so forth. During instruction, multiple pedagogical methods, tools, and activities are employed to meet the learners' needs (Bermel 2016). This enables learners to become more motivated and productive (Tahiri, Bennani, and Khalidi Idrissi 2017).

In computing education, students possess varying levels of access to technology, and some of them may have disabilities in fully participating in computing activities. For example, students may have difficulty using the mouse such as not knowing how to left-click, double click, or drag an object. They may also face problems when using the keyboard functions such as "Control-Alt-Delete." Not only that, but they may also fail to read or interpret commands of Scratch and Etoys. Therefore, the

instructors have to utilize differentiated instruction including peer support, modeling, and scaffolding (Israel et al. 2015) to accommodate these struggling students (Hansen et al. 2016). Through differentiated instruction, students can enter at their level and work to their potential in the computing environment (Gadanidis and Caswell 2018).

According to Rajala et al. (2016), formal learning is regarded as planned through academic institutions and leading to standard qualifications. Voogt et al. (2015) claim that CT was positioned in the formal curriculum in two major ways, namely computing as a separate subject and CT in cross-curricular practices. It was argued by The Royal Society (2012) that "every child should have the opportunity to learn concepts and principles from Computing (including Computer Science and Information Technology) from the beginning of primary education onwards, and by age 14 should be able to choose to study toward a recognized qualification in these areas" (44).

Informal learning is viewed as "what happens outside the structures and boundaries of formal education, the topic or focus of which is determined by the person doing the learning, on their own or with others" (Davies and Eynon 2013, 330). Wing (2008) contended that CT should not just involve formal learning but also informal learning as "learning takes place in many ways and outside the classroom: children teach each other; learn from parents and family; learn at home, in museums and in libraries; and learn through hobbies, surfing the Web and life experiences" (3721). Unlike formal learning, informal is tangible, open-ended, interest- and practice-driven, and highly contextualized (Arnesen et al. 2016). Informal learning is not restricted by school culture (Wong, Jan, and Liang 2014) as it occurs outside of prearranged lessons (McCartney et al. 2010). It is conceived as learning that has less structure, intrinsic motivation, and fewer formal assessments, and learning outcomes are not evaluated explicitly. It is performed spontaneously and not always prearranged (Eshach 2006). Informal learning is voluntary, usually learner-led, and nonsequential as compared to the rather compulsory, teacher-led, and sequential formal learning (Eshach 2006). In both formal and informal CT settings, the nature of the CT tasks and the supports and scaffolding given by the adults will influence the engagement of the learners in different levels of CT skills (Ehsan et al. 2018).

Leveraging computational tools has generated new potentials for linking learning taking place in different venues, linking people with mutual expertise and interests, and for incorporating informal learning within formal learning (Laru and Jarvela 2015). Digital practices in-school and out-of-school learning among the students is not separate, but it provides a supportive ecosystem for a varied range of students by making learning more engaging (Kumpulainen and Mikkola 2016).

The informal learning approaches introduced through inquiry-based, project-based, and problem-based learning techniques using technology would shift control from teachers to students (Schuck et al. 2017). Furthermore, informal practices and skills could be enhanced through digital making, mobile learning, social media, gaming, and getting involved in online communities. This would support the schoolwork in the formal classroom (Erstad, Gilje, and Arnseth 2013). In other words, daily digital practices can complement formal learning among young people.

Despite the disadvantages of informal education, it is believed that informal education can assist in reshaping formal education (Lewin and Charania 2018) by shifting the formal learning practices from transmissive techniques to student-centered, self-directed, and collaborative methods to deepen the knowledge of the students (Khaddage et al. 2016). A study was done by Boyle and McDougall (2003) to explore the formal and informal learning environments for the teaching and learning of computer programming. They found that although most of the learners are classified as formal learners with more external sources of control, informal learners show deep technical proficiency and various finished products, which shows that it may be best to leave some learners to obtain their own programming skills.

Boustedt et al. (2011) discovered many advantages of informal learning of computing topics from the computer science students including (a) the learner can select the level of study, either specific or conceptual; (b) a sense of achievement in informal learning, such as self-confidence and satisfaction; (c) things learned informally can usually be better engaged; (d) informal learning can be more flexible according to time and pace that suit the learner; (e) informal learning is motivating and can make topics interesting; (f) informal learning can be employed to increase the breadth or depth of learning in formal courses; and (g) informal learning can promote formal

learning and vice versa. However, there are difficulties with informal learning, including (a) students may miss crucial features of the topic in informal learning; (b) informal learning is usually ad hoc and stops when the task can be solved using the knowledge gained so far; (c) some students need the structure and deadline provided by the teacher; (d) for learners, informal learning may be difficult to evaluate how learners know when they have learned enough knowledge; (e) without specific deadlines and rewards, less time may be spent on informal learning; (f) informal learners may miss chances for classroom discussions and other social interactions that are crucial for learning.

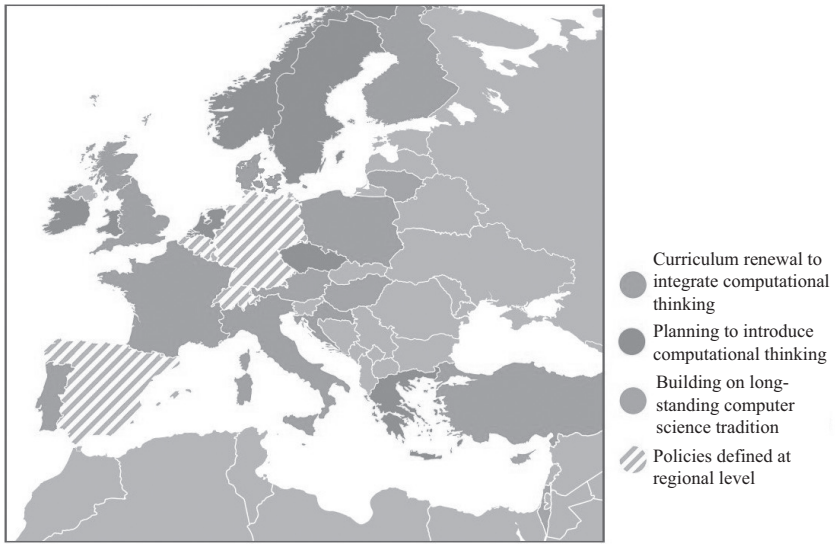
COMPUTATIONAL THINKING IN THE FORMAL CURRICULUM

There are two major trends seen in the manner in which various countries incorporate CT in the formal curriculum. The first trend is enhancing CT skills in children and young people to allow them to think differently, solve real-world problems, express themselves via various media, and investigate daily matters from a different viewpoint. The second trend is facilitating CT to augment the economic evolution, fill job openings in information and communication technologies, and prepare for a future career (Bocconi et al. 2016). The countries that have included CT into the formal curriculum can be classified into three clusters:

- cluster I: the ongoing curriculum reform process;
- cluster II: planning to introduce CT; and
- cluster III: building on a long-standing computer science education tradition.

Some countries have developed CT curriculum or policy initiatives at the regional level as shown in figure 7.1 (Bocconi et al. 2016). CT is positioned in the curriculum in terms of two criteria: (1) education level and (2) subject level.

Wing (2008) claimed that CT should not only be taught in the university but also incorporated into levels spanning primary to secondary. At the education level, countries are mostly seen to immerse CT at the secondary school level, but recently an emerging trend toward incorporation of CT at the primary school level has been seen as well. In some countries,



7.1 Integrating CT in the formal curriculum for European countries (Bocconi et al. 2016).

CT can be taught as a separate subject, and in others, it is embedded into subjects in the school syllabus (Voogt et al. 2015).

Among cluster I countries, Estonia introduced CT into the formal curriculum by launching the ProgeTiger program in 2012. This program intends to integrate robotics and programming into primary and secondary education. There are three thematic areas that are combined under this program, namely design and technology, ICT, and engineering sciences (Heintz, Mannila, and Färnqvist 2016). In Slovakia, coding is integrated as a compulsory component at all levels of school education. Therefore, all the students will learn coding throughout school (Balanskat and Engelhardt 2015).

England is one of the European countries that has introduced the computing subject in primary and secondary schools to replace the existing ICT curricula since 2014 (Bocconi et al. 2016). There are four key stages of K–12 education in England where students are anticipated to foster CT skills gradually, including key stage 1 (ages five to seven), key stage 2 (ages seven to eleven), key stage 3 (ages eleven to fourteen), and key stage 4 (ages fourteen to sixteen) (Department for Education 2019).

Another cluster I country is Finland. In 2016, Finland set up a new national curriculum for primary and secondary education (Mannila et al.

2014). This new curriculum includes programming as an integrated component in primary education but not for secondary education (Heintz, Mannila, and Färnqvist 2016). Poland is another country under cluster I, where a new computer science (informatics) curriculum has been implemented in the K–12 formal schools in 2017. The major aim of this curriculum is to encourage students to utilize CT in solving problems for a variety of subjects (Bocconi et al. 2016). In Australia, digital technologies and design and technologies are introduced as computing subjects to enhance students' CT skills. These subjects have been made compulsory for primary and secondary students to learn (Bocconi et al. 2016; Heintz, Mannila, and Färnqvist 2016). Bulgaria and Hungary integrate coding into the subject of informatics at the secondary school levels (Balanskat and Engelhardt 2015).

Regarding cluster II, Norway has not yet introduced computer science into their K–12 syllabuses, but they have a large pilot program that introduces programming as an elective subject at the secondary levels in 2016 (Heintz, Mannila, and Färnqvist 2016). In 2015, Sweden established a new curriculum for primary levels and also reorganized their secondary curriculum to reinforce the programming and digital competence. IT-related subjects are compulsory for primary students, but they are elective subjects for secondary students (Mannila et al. 2014).

For South Korea, a compulsory subject called "informatics" is introduced to the primary schools, and it is an elective subject in secondary schools (Heintz, Mannila, and Färnqvist 2016). Netherlands secondary schools have a computer science subject, but it is not a compulsory subject. The schools can decide whether to teach this subject to the students, and the students have the right to select this subject. In Ireland, an optional short course of coding has been introduced for the junior cycle program at the secondary school level, but there is no national program at the primary school level (Balanskat and Engelhardt 2015).

In cluster III, there are some countries that incorporated CT into formal education by building on a long-standing computer science education tradition. Israel is one of the countries that have had computer science education for a very long time at the secondary school level. Their computer science course aims to enhance students' algorithmic and logical thinking (Bocconi et al. 2016). Lithuania has a compulsory subject called "informatics" that has been revised to become "information

technology" in secondary schools (Mannila et al. 2014). The five areas covered in information technology are digital technologies, virtual communication, information, programming, and algorithms in addition to ethics, security, and legal principles (Bocconi et al. 2016).

Several countries have developed CT curriculum or policy initiatives at the regional level. For instance, the state of Bavaria, Germany has had its own compulsory subject of computer science at the secondary levels since 2014. Digital competence, a programming-related subject, is mandatory for primary schools in one of the regions of Spain, Catalonia (Bocconi et al. 2016). Table 7.1 reveals the overview of computing education in different countries (adapted from Heintz, Mannila, and Färnqvist 2016).

COMPUTATIONAL THINKING IN THE INFORMAL CURRICULUM

According to Wing (2008), informal learning of CT ought to be explored as it happens anytime and everywhere. The students teach each other, learn from their family members, and learn in the libraries, in museums, and at home as well as through surfing websites and hobbies. Meetups like Codo Dojos, where students can learn programming from more experienced coders, and Code Clubs in after-school programs provide opportunities for learning in an informal environment. Fishman and Dede (2016) stated that the students can learn CT informally by engaging themselves as creators and makers, such as Do It Yourself digital textiles, Scratch programming, and robotics competitions. A study was done by McCartney et al. (2010) to determine how computing students learn computing topics informally. The findings revealed that the topics selected by the students were programming languages (e.g., C++ and Visual Basic), hardware (e.g., wiring two circuit boards), and software (e.g., Netbeans). These students chose to learn these topics because they desired to complete some projects. The resources utilized by the students were the Internet and thesauruses or books. Meanwhile, the learning approaches were searching for good examples, working with more knowledgeable mentors and friends, trial and error, talking to and observing others, and so on. The success of the projects was exploited to evaluate their learning. Overall, the informal learning setting brought a positive impact on the students' learning among computing students.

Table 7.1 Overview of computing education in different countries

Cluster	Country	Subject	Method	Primary	Secondary
I	Estonia	Programming (technology and innovation)	Integrated	Compulsory	Compulsory
I	Slovakia	Coding	Integrated	Compulsory	Compulsory
I	England	Computing	Replace existing subject	Compulsory	–
I	Finland	Programming (digital competence)	Integrated	Compulsory	–
I	Poland	Computer science (informatics)	Own subject	Compulsory	Compulsory
I	Australia	Digital technologies and design and technologies	Own subject and integrated	Compulsory	Compulsory
I	Bulgaria	Informatics	Integrated	–	Compulsory
I	Hungary	Informatics	Integrated	–	Compulsory
II	Norway	Programming	Own subject	–	Elective
II	Sweden	Programming and digital competence	Integrated	Compulsory	Elective
II	South Korea	Informatics	Own subject	Compulsory	Elective
II	Netherlands	Computer science	Own subject	–	Elective
II	Ireland	Coding (short course)	Own subject	–	Elective
III	Israel	Computer science	Own subject	–	Elective
III	Lithuania	Information technology	Revise existing subject	–	Compulsory

Lee et al. (2011) performed a study to enhance students' CT skills by conducting out-of-school programs through modeling and simulation, robotics, and game design and development. They engaged the youth in the rich CT environments using a three-stage progression, which was a use-modify-create framework. In the use stage, the students were required to use someone's creation, for instance, play a readymade computer game and run a program to control a robot. After that, the students would start

to modify the game and program it to make it more sophisticated in the modify stage. Eventually, the students would create original products by employing their rising CT skills: automation, analysis, and abstraction.

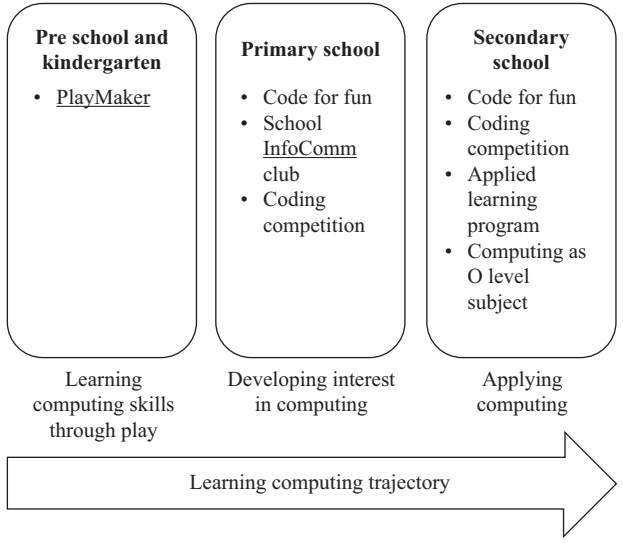
Other work conducted by Ahmadi, Jazayeri, and Kandoni (2012) studied novice programmers in solving CT problems during the game design procedures in an informal learning context. AgentWeb, a cloud-based, online game design environment was utilized to allow the learners to construct their games beyond the formal classrooms. The learners' CT skills could be reinforced while identifying the game object, examining the behavior of the game object, and programming the behavior. Furthermore, Boulton et al. (2016) implemented a study to explore the role of game jams in stimulating CT among the learners aged eleven to eighteen in the informal learning setting. Game jams is an interesting approach that encourages a small game creation in a short time, with an aim to engage learners in learning computing and promoting their CT skills. The findings exhibited that the application of the mobile app, Pocket Code, in the game jams motivated the learners in the learning process.

Mesiti et al. (2019) researched how to develop CT capacity among youth through an informal learning setting in the Museum of Science, Boston. Three specific designs to convey CT content in the Pixar exhibition were utilized. Three dimensions of CT were focused on, namely concepts (i.e., patterns, decomposition, abstraction, and algorithms), practices (i.e., collaborating, creating, and debugging), and perspectives (i.e., empowered to ask questions about technology, sense of identity, and relationship to the technical world). The project research was accomplished in two phases. In phase 1, the focus was on how the learners could be reinforced to interact with exhibits and comprehend problem-solving approaches that address creative and complex challenges in computer programming. Phase 2 focused on exploring the affordances of these exhibits to develop CT capacity, interest, and feelings of efficiency in problem decomposition. The results displayed that the interest of youth in learning programming was augmented and their problem decomposition perception, creativity perception, and self-efficacy in computer programming became enhanced after the usage of exhibits. Not only that; they also understood the value and significance of mathematics and programming.

COMPUTING EDUCATION IN SINGAPORE

In 2014, Singapore launched the Smart Nation program, a nationwide effort to harness technology in sectors of business, government, and home to improve urban living, build stronger communities, grow the economy, and create opportunities for all residents to address the ever-changing global challenges (Smart Nation 2017). To support the Smart Nation initiative, one of the key enablers is to develop computational capabilities. Programs were implemented to introduce and develop CT skills and coding capabilities from preschool children to adults. We will next describe the landscape of K–12 CT- and coding-related programs in Singapore that are implemented by the schools and various government organizations.

Unlike countries like Finland, England, and South Korea, Singapore has not included computing or CT as compulsory education. However, learning coding has made inroads into schools initially through voluntary participation. Singapore's approach has been to provide opportunities for students to develop their interests in coding and computing skills through various touchpoints at various ages as shown in figure 7.2. Computing and CT skills are introduced to the children that are age-appropriate and intended to engage and stimulate their interests. For preschool and

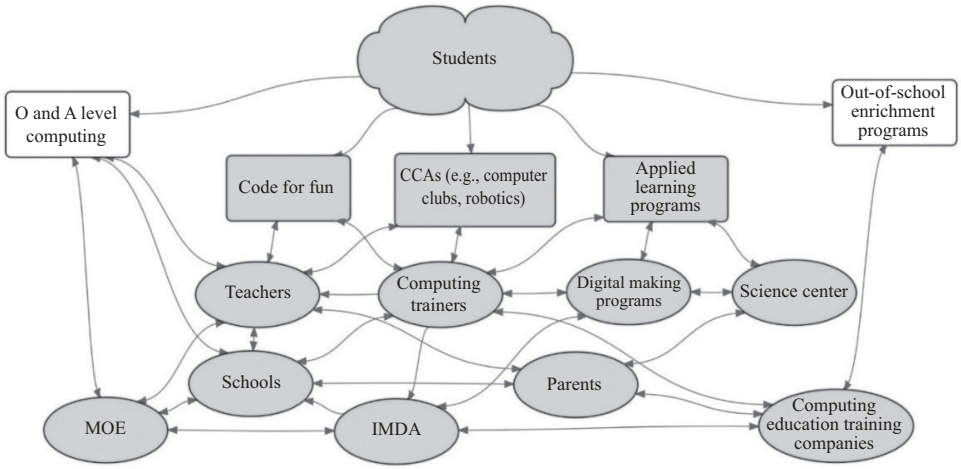


7.2 Learning pathways for computing education.

kindergarten children, the Playmaker program was introduced by the InfoComm and Digital Media Authority (IMDA), in collaboration with Temasek Polytechnic's Early Childhood Centre, to Singapore preschools (IMDA 2017). Teachers can use selected e-toys such as the BeeBot and Kibo to develop CT skills such as problem-solving and algorithmic thinking through play.

Many primary schools in Singapore opt-in to offer students the ten-hour or twenty-hour Code for Fun program conducted via the Ministry of Education (MOE) or by selected IMDA technology training partners where they are introduced to coding in Scratch and programming hardware such as Microbit or robotics kits. Lately in 2019, Code for Fun has been made compulsory for ten instructional hours during the upper primary years, that is, for grades 5 and 6 (Straits Times 2019). Secondary schools offer students opportunities to learn coding and develop CT skills through the Code for Fun, Digital Maker, and Applied Learning Programmes (ALP). As informal learning programs, co-curricular activities (CCAs) such as info-communications clubs in primary and secondary schools offer students opportunities to learn and apply computational skills. In the formal school curricula, computing is offered as a subject for grades 9 and 10 in the "O" levels, and computer science for grades 11 and 12 in the "A" levels. Thus, for students who take up these subjects, it becomes formal learning for them.

By incorporating both informal learning pursuits as well as formal learning courses, the programs provide varied experiences for our students to develop their own interests and apply computing to solving problems. The programs are facilitated, developed, and implemented by stakeholders in the ecosystem shown in figure 7.3. The agents in the ecosystem directly involved with students include teachers, school leaders, and parents. External agents include the government agencies such as the Infocomm Media Development Authority, Ministry of Education, Science Centre, and other computing educators. The range of programs for computing includes formal computing education in the O and A level of computing; informal programs in schools such as Code for Fun, Play/Digital Maker, computing-based CCAs, and the Applied Learning program; and the out-of-school enrichment programs such as the ones offered by computing educational centers or the Science Centre.



7.3 The computing education ecosystem in Singapore.

DISCUSSION AND CONCLUSION

In a nutshell, formal and informal systems play an important role as differentiated learning pathways in computing education. Both systems can be utilized to accommodate students of different levels and develop their CT skills in K–12 schools. The potential of informal learning to complement the formal learning about computing has been evidenced in earlier studies, such as Erstad, Gilje, and Arnseth (2013); Peeters et al. (2014); Guzdial et al. (2014); and Lewin and Charania (2018). However, the challenging part for informal learning of computing education is that it is not recognized as legitimate learning because it is implemented outside the school (Peeters et al. 2014). Also, immersing informal learning practices in the formal classroom remains problematic as a result of discrepancies with assessment practices, lack of technology infrastructure, and timetable limitations (Lewin and Charania 2018). Inadequate funds can also restrict out-of-school resources such as Internet and technology access, thus leading to inconsistencies in informal learning practices (Lopez and Caspe 2014).

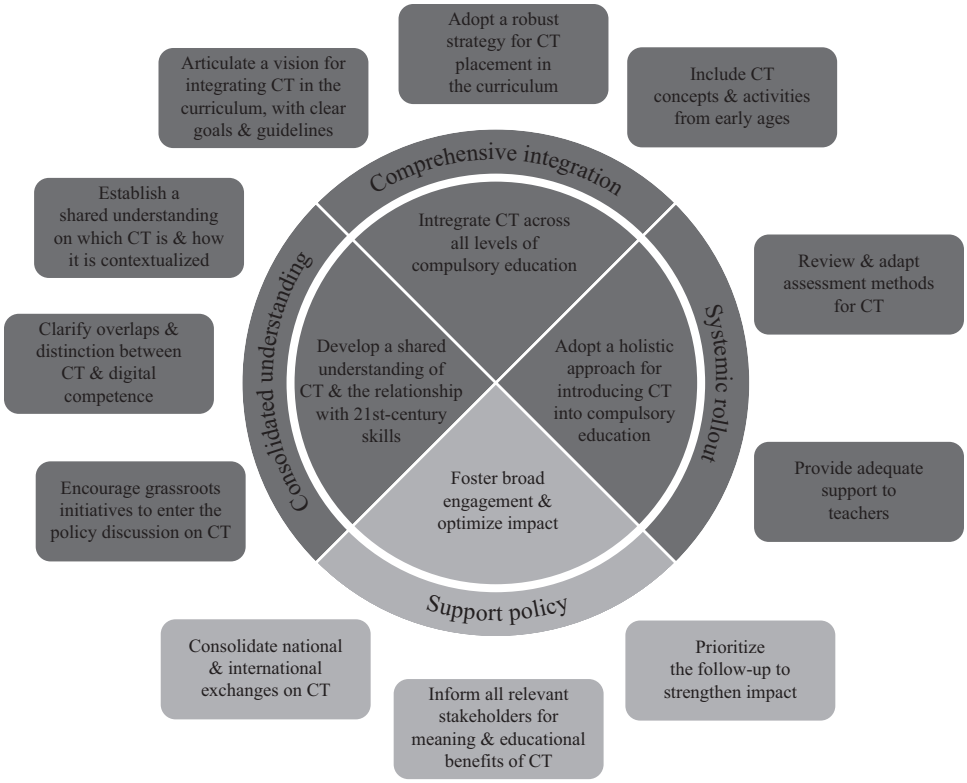
A key aspect of addressing these challenges is professional development for teachers to implement CT in the classrooms. The computing or computer science teachers have to connect the core concepts of CT to other domains. Meanwhile, teachers from other domains should be familiar

with the core concepts of CT (Voogt et al. 2015). The teachers need to be trained through professional development programs to fully harness informal learning to support the attainment of formal learning in computing education. They should provide more guidance and support to the students in connecting in-school and out-of-school learning for knowledge construction (Lewin and Charania 2018).

Hence, the suggestion is to infuse CT into the formal curriculum to make sure all children have equal chances to be equipped with the CT skills they require in a digital world. Not only that, but the teaching of CT should begin from early childhood as this can be a rewarding and engaging experience for young learners (Bers 2008). In earlier studies (e.g., Wyeth 2008; Kazakoff, Sullivan, and Bers 2012), children as young as four to six years old were able to construct and program simple robotics projects as well as build their CT skills and learn powerful ideas from computer programming, technology, and engineering (Bers 2008). Therefore, the government and policymakers ought to establish their goal and vision as well as judiciously describe, design, and observe their concrete execution actions. Setting particular targets is essential not only to notifying concrete execution options but also to getting related stakeholders on board (Bocconi et al. 2016). Thus, Bocconi et al. (2016) propose the policy and practice implications for introducing CT in formal education as shown in figure 7.4.

In figure 7.4, the policy and practice implications consist of various phrases (i.e., from preparation to execution), and they are all interconnected robustly. Four main parts should be the focus of policymakers and stakeholders: (1) consolidated CT understanding; (2) comprehensive integration; (3) systematic rollout; and (4) support policy. The first part, consolidating CT understanding, aims to promote a shared comprehension of CT and its relationship with twenty-first-century skills. It includes launching a shared comprehension on what CT is and how it is contextualized, elucidating differences and overlaps between CT and digital competence, and boosting grassroots initiatives to get involved in the policy discussion on CT.

The second part on comprehensive integration has three phases to infuse CT across all education levels: that is, enunciating a vision for



7.4 Policy and practice implications for introducing CT in formal education (Bocconi et al. 2016).

incorporating CT in the curriculum, utilizing a strong approach for CT positioning in the curriculum, and involving CT concepts and activities from early ages. Furthermore, there are two phases for systemic rollout to implement a holistic tactic for introducing CT, which are evaluating and modifying assessment techniques for CT, and giving sufficient support to teachers. Support policy, the fourth part of this framework, is intended to encourage extensive engagement and strengthen the effect of infusing CT across all education levels by combining national and international exchanges on CT, notifying all related stakeholders of the educational advantages and the meaning of CT, and prioritizing the follow-up to reinforce the impact (Bocconi et al. 2016).

ACKNOWLEDGMENTS

The work reported in this book chapter is supported by a grant from the Ministry of Education's Educational Research Funding Programme (OER 10/18 LCK).

REFERENCES

Ahmadi, Navid, Mehdi Jazayeri, and Monica Landoni. 2012. "Helping Novice Programmers to Bootstrap in the Cloud: Incorporating Support for Computational Thinking into the Game Design Process." In *Proceedings of the 2012 IEEE 12th International Conference on Advanced Learning Technologies, ICALT '12*, 349–353. Washington, DC: IEEE Computer Society.

Alves, Nathalia Da Cruz, Von Wangenheim, Christiane Gresse, and Jean C. R. Hauck. 2019. "Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study." *Informatics in Education* 18, no. 1: 17–39.

Arnesen, Thomas, Eyvind Elstad, Gavriel Salomon, and Lars Vavik. 2016. "Educational Technology and Polycontextual Bridging: An Introduction." In *Educational Technology and Polycontextual Bridging*, edited by Eyvind Elstad, 3–14. Rotterdam/Boston/Taipei: Sense Publishers.

Balanskat, Anja, and Katja Engelhardt. 2015. *Computing Our Future: Computer Programming and Coding—Priorities, School Curricula and Initiatives across Europe*. Belgium: European Schoolnet.

Bermel, Jodi A. 2016. *Using One-to-one Computing for Differentiated Instruction in Iowa: An Investigation of the Impact of Teachers' Perceptions of Teaching and Learning*. Electronic Theses and Dissertations. 342.

Bers, Marina Umaschi. 2008. *Blocks, Robots and Computers: Learning about Technology in Early Childhood*. New York: Teacher's College Press.

Bocconi, Stefania, Augusto Chioccariello, Giuliana Dettori, Anusca Ferrari, and Katja Engelhardt. 2016. *Developing Computational Thinking in Compulsory Education*. Luxembourg: Publications Office of the European Union.

Boulton, Helen, Bernadette Spieler, Anja Petri, Christian Schindler, Wolfgang Slany, and Maria Beltran. 2016. "The Role of Game Jams in Developing Informal Learning of Computational Thinking: A Cross-European Case Study." In *Proceedings of EDULEARN16*, 4–6 July 2016, Barcelona, Spain, 7034–7044.

Boustedt, Jones, Anna Eckerdal, Robert McCartney, Kate Sanders, Lynda Thomas, and Carol Zander. 2011. "Students' Perceptions of the Differences between Formal and Informal Learning." In *Proceedings of the Seventh International Workshop on Computing Education Research*, 61–68. New York: ACM.

Boyle, Martin, and Anne McDougall. 2003. *Formal and Informal Environments for the Learning and Teaching of Computer Programming*. Paper presented at the IFIP Working

Groups 3.1 and 3.3 Working Conference: ICT and the Teacher of the Future, St. Hilda's College, The University of Melbourne, Australia 27–31 January 2003.

Burleson, Winslow S., Danielle B. Harlow, Katherine. J. Nilsen, Ken Perlin, Natalie Freed, Camilla Norgaard Jensen, Byron Lahey, Patrick Lu, and Kasia Muldner. 2018. "Active Learning Environments with Robotic Tangibles: Children's Physical and Virtual Spatial Programming Experiences." *IEEE Transactions on Learning Technologies* 11, no. 1: 96–106.

Computer Science Teachers Association (CSTA). 2016. "K–12 Computer Science Framework." Accessed October 2, 2019. http://www.k12cs.org.

Dabbagh, Nada, and Anastasia Kitsantas. 2012. "Personal Learning Environments, Social Media, and Self-Regulated Learning: A Natural Formula for Connecting Formal and Informal Learning." *Internet and Higher Education* 15, no. 1: 3–8.

Davies, Chris, and Rebecca Eynon. 2013. "Studies of the Internet in Learning and Education: Broadening the Disciplinary Landscape of Research." In *The Oxford Handbook of Internet Studies*, edited by William H. Dutton, 328–349. Oxford: Oxford University Press.

Department for Education. 2019. "National Curriculum in England: Computing Programmes of Study." Accessed October 3, 2019. https://www.gov.uk/government/publications/national-curriculum-in-englandcomputing-programmes-of-study/national-curriculum-in-england-computing-programmes-ofstudy.

Ehsan, Hoda, Tikyna Dandridge, Ibrahim H. Yeter, and Monica E. Cardella. 2018. "K-2 Students' Computational Thinking Engagement in Formal and Informal Learning Settings: A Case Study." In *Proceedings of the American Society for Engineering Education (ASEE) Conference & Exposition*. Salt Lake City, UT.

Erstad, Ola, and Julian Sefton-Green. 2013. "Digital Disconnect? The 'Digital Learner' and the School." In *Identity, Community, and Learning Lives in the Digital Age*, edited by Ola Erstad and Julian Sefton-Green, 87–104. New York: Cambridge University Press.

Erstad, Ola, Øystein Gilje, and Hans Christian Arnseth. 2013. "Learning Lives Connected: Digital Youth across School and Community Spaces." *Comunicar* 40: 89–98.

Eshach, Haim. 2006. "Bridging In-School and Out-Of-School Learning: Formal, Non-Formal, and Informal." In *Science Literacy in Primary Schools and Pre-Schools*. *Classics in Science Education* (vol. 1), edited by Haim Eshach, 115–141. Springer, Dordrecht. https://doi.org/10.1007/1-4020-4674-X_5.

Fields, Deborah. A., Michael Giang, and Yasmin Kafai. 2014. "Programming in the Wild: Trends in Youth Computational Participation in the Online Scratch Community." In *Proceedings of the 9th Workshop in Primary and Secondary Computing Education, WiPSCE '14*, 2–11, New York.

Fishman, Barry, and Chris Dede. 2016. "Teaching and Technology: New Tools for New Times." In *Handbook of Research on Teaching, 5th Edition (American Educational Research Association)*, edited by Drew H. Gitomer and Courtney A. Bell. New York: Springer.

Gadanidis, George, and Beverly Caswell. 2018. *Computational Modelling in Elementary Mathematics Education: Making Sense of Coding in Elementary Classrooms*. KNAER Mathematics Knowledge Network White Paper.

Guzdial, Mark. 2015. *Learner-Centered Design of Computing Education: Research on Computing for Everyone*. Morgan-Claypool.

Guzdial, Mark, Barbara Ericson, Tom Mcklin, and Shelly Engelman. 2014. "Georgia Computes! An Intervention in a US State, with Formal and Informal Education in a Policy Context." *ACM Transactions on Computing Education* 14, 13:1–13:29.

Hansen, Alexandria. K., Eric R. Hansen, Hilary A. Dwyer, Danielle B. Harlow, and Diana Franklin. 2016. "Differentiating for Diversity: Using Universal Design for Learning in Computer Science Education." In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education—SIGCSE '16*, February 2016, 376–381.

Heintz, Fredrik, Linda Mannila, and Tommy Färnqvist. 2016. "A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K-12 Education." *IEEE Frontiers in Education Conference (FIE)*, 1–9.

Ho, Andrew, Isaac Chuang, Justin Reich, Cody Coleman, Jacob Whitehill, Curtis Northcutt, Joseph Williams, John Hansen, Glenn Lopez, and Rebecca Petersen. 2015. "HarvardX and MITx: Two Years of Open Online Courses Fall 2012–Summer 2014." Accessed October 2, 2019. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2586847.

Hynes, Morgan M., Monica E. Cardella, Tamara J. Moore, Sean P. Brophy, Senay Purzer, Kristina Maruyama Tank, Muhsin Menekse, Ibrahim H. Yeter, and Huda Ehsan. 2019. "Inspiring Young Children to Engage in Computational Thinking In and Out of School." In *Proceeding of American Society for Engineering Education (ASEE) Conference & Exposition*. Tampa, FL.

IMDA. 2017. "PlayMaker Changing the Game." Accessed October 15, 2019. https://www.imda.gov.sg/infocomm-and-media-news/buzz-central/2015/10/playmaker-changing-the-game.

Israel, Maya, Jamie N. Pearson, Tanya Tapia, Quentin M. Wherfel, and George Reese. 2015. "Supporting All Learners in School-Wide Computational Thinking: A Cross-Case Qualitative Analysis." *Computers & Education* 82: 263–279.

Kazakoff, Elizabeth R., Amanda Sullivan, and Marina U. Bers. 2012. "The Effect of a Classroom-Based Intensive Robotics and Programming Workshop on Sequencing Ability in Early Childhood." *Early Childhood Education Journal* 41, no. 4: 245–255.

Khaddage, Ferial, Wolfgang Müller, and Kim Flintoff. 2016. "Advancing Mobile Learning in Formal and Informal Settings via Mobile App Technology: Where to from Here, and How?" *Educational Technology & Society* 19, 3: 16–26.

Kjällander, Susanne, Anna Åkerfeldt, Linda Mannila, and Peter Parnes. 2018. "Makerspaces Across Settings: Didactic Design for Programming in Formal and Informal Teacher Education in the Nordic Countries." *Journal of Digital Learning in Teacher Education* 34, no. 1: 18–30.

Kumpulainen, Kristiina, and Anna Mikkola. 2016. "Toward Hybrid Learning: Educational Engagement and Learning in the Digital Age." In *Educational Technology and Polycontextual Bridging*, edited by Eyvind Elstad, 15–38. Rotterdam/Boston/Taipei: Sense Publishers.

Laru, Jari, and Sanna Järvelä. 2015. "Integrated Use of Multiple Social Software Tools and Face-to-Face Activities to Support Self-regulated Learning: A Case Study in a Higher Education Context." In *Seamless Learning in the Age of Mobile Connectivity*, edited by Lung-Hsiang Wong, Milrad Marcelo, and Specht Marcus, 471–484. Singapore: Springer.

Lee, Irene, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. "Computational Thinking for Youth in Practice." *ACM Inroads* 2, no. 1: 33–37.

Lopez, M. Elena, and Margaret Caspe. 2014. *Family Engagement in Anywhere, Anytime Learning*. Cambridge, MA: Harvard Family Research Project.

Lunenburg, Fred C. 2010. Extracurricular Activities. *Schooling* 1, no. 1: 1–4.

Manches, Andrew, and Lydia Plowman. 2017. "Computing Education in Children's Early Years: A Call for Debate." *British Journal of Educational Technology* 48, no. 1: 191–201.

Mannila, Linda, Valentina Dagiene, Barbara Demo, Natasa Grgurina, Claudio Mirolo, Lennart Rolandsson, and Amber Settle. 2014. "Computational Thinking in K-9 Education." In *Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference, ITiCSE-WGR 2014*, 1–29. New York: ACM.

McCartney, Robert, Anna Eckerdal, Jam Erik Moström, Kate Sanders, Lynda Thomas, and Carol Zander. 2010. "Computing Students Learning Computing Informally." In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research—Koli Calling '10*, 43–8.

Mesiti, Leigh Ann, Alana Parkes, Sunewan C. Paneto, and Clara Cahill. 2019. "Building Capacity for Computational Thinking in Youth through Informal Education." *Journal of Museum Education* 44, no. 1: 108–21.

Peeters, Jeltsen, Free De Backer, Tine Buffel, Ankelien Kindekens, Katrien Struyven, Chang Zhu, and Koen Lombaerts. 2014. "Adult Learners' Informal Learning Experiences in Formal Education Setting." *Journal of Adult Development* 21, no. 3: 181–192.

Pereira, Roberto, M. Cecilia C. Baranauskas, and Sergio Roberto P. da Silva. 2013. "Social Software and Educational Technology: Informal, Formal and Technical Values." *Educational Technology & Society* 16, 1: 4–14.

Rajala, Antti, Kristiina Kumpulainen, Jaakko Hilppö, Maiju Paananen, and Lasse Lipponen. 2016. "Connecting Learning across School and Out-of-School Contexts: A Review of Pedagogical Approaches." In *Learning across Contexts in the Knowledge Society*, edited by Ola Erstad, Kristiina Kumpulainen, Asa Makitalo, Kim Christian

Schroder, Pille Pruulmann-Vengerfeldt, and Thuridur Johannsdottir, 15–38. Rotterdam/Boston/Taipei: Sense Publishers.

Roman-Gonzalez, Marcos, Juan-Carlos C. Perez-Gonzalez, Jesus Moreno-Leon, and Gregorio Robles. 2018. "Can Computational Talent Be Detected? Predictive Validity of the Computational Thinking Test." *International Journal of Child-Computer Interaction* 18: 47–58.

Schuck, Sandy, Matthew Kearney, and Kevin Burden. 2017. "Exploring Mobile Learning in the Third Space." *Technology, Pedagogy and Education* 26, no. 2: 121–137.

Smart Nation. 2017. "Why Smart Nation." Accessed October 10, 2019. https://www.smartnation.sg/about-smart-nation.

Snodgrass, Melinda R., Maya Israel, and George C. Reese. 2016. "Instructional Supports for Students with Disabilities in K-5 Computing: Findings from a Cross-Case Analysis." *Computers & Education* 100: 1–17.

Strait Times (2019). "Coding Classes for All Upper Primary Pupils from 2020." Accessed July 12, 2019. https://www.straitstimes.com/tech/coding-classes-for-all-upper-primary-pupils-from-2020.

Tahiri, Jihane Sophia, Samir Bennani, and Mohamed Khalidi Idrissi. 2017. "diffMOOC: Differentiated Learning Paths Through the Use of Differentiated Instruction within MOOC." *International Journal of Emerging Technologies in Learning (iJET)* 12, no. 03: 197–218.

The Royal Society. 2012. "Shut Down or Restart? The Way Forward for Computing in UK Schools." London: The Royal Society. Accessed October 9, 2019. http://royalsociety.org/uploadedFiles/Royal_Society_Content/education/policy/computing-in-schools/2012-01-12-Computing-in-Schools.pdf.

Voogt, Joke, Petra Fisser, Jon Good, Punya Mishra, and Aman Yadav. 2015. "Computational Thinking in Compulsory Education: Towards an Agenda for Research and Practice." *Education and Information Technologies* 20, no. 4: 715–728.

Wing, Jeannette M. 2006. "Computational Thinking." *Communications of the ACM* 49, no. 3: 33–36.

Wing, Jeannette. 2008. "Computational Thinking and Thinking about Computing." *Philosophical Transactions of the Royal Society* 366, no. 1881: 3717–3725.

Wong, Lung Hsiang, Mingfong Jan, and Rose Liang. 2014. "Learning Sciences". In NIE Working Paper Series No. 1, edited by Wing On Lee and David Hung, 5–41. Office of Education Research, National Institute of Education, Nanyang Technological University.

Wyeth, Peta. 2008. "How Young Children Learn to Program with Sensor, Action, and Logic Blocks." *International Journal of the Learning Sciences* 17, no. 4: 517–550.

This is a section of doi:10.7551/mitpress/14041.001.0001

Computational Thinking Curricula in K–12

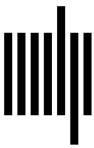
International Implementations

Edited by: Harold Abelson, Siu-Cheung Kong

Citation:

Computational Thinking Curricula in K–12: International Implementations
Edited by: Harold Abelson, Siu-Cheung Kong
DOI: 10.7551/mitpress/14041.001.0001
ISBN (electronic): 9780262378642
Publisher: The MIT Press
Published: 2024

The open access edition of this book was made possible by generous funding and support from MIT Press Direct to Open



The MIT Press

The MIT Press would like to thank the anonymous peer reviewers who provided comments on drafts of this book. The generous work of academic experts is essential for establishing the authority and quality of our publications. We acknowledge with gratitude the contributions of these otherwise uncredited readers.

This book was set in Stone Serif by Westchester Publishing Services.

Library of Congress Cataloging-in-Publication Data

Names: Abelson, Harold, editor. | Kong, Siu Cheung, editor.
Title: Computational thinking curricula in K-12 : international implementations / edited by Harold Abelson and Siu-Cheung Kong.
Description: Cambridge, Massachusetts : The MIT Press, [2024] | Includes bibliographical references and index.
Identifiers: LCCN 2023027971 (print) | LCCN 2023027972 (ebook) | ISBN 9780262548052 (paperback) | ISBN 9780262378659 (epub) | ISBN 9780262378642 (pdf)
Subjects: LCSH: Computer science—Study and teaching—Case studies. | Problem solving—Study and teaching—Case studies.
Classification: LCC QA76.27 .C478 2024 (print) | LCC QA76.27 (ebook) | DDC 004.071—dc23/eng/20230905
LC record available at https://lccn.loc.gov/2023027971
LC ebook record available at https://lccn.loc.gov/2023027972

ISBN: 978-0-262-54805-2