

Chapter 19

Educational Policy and Implementation of Computational Thinking and Programming: Case Study of Singapore



Peter Seow, Chee-Kit Looi, Meng-Leong How, Bimlesh Wadhwa and Long-Kai Wu

Abstract Many countries that recognise the importance of Computational Thinking (CT) skills are implementing curriculum changes to integrate the development of these skills and to introduce programming into formal school education. In countries such as the United Kingdom, Lithuania, Finland, Korea and Japan, initiatives and policies are made to introduce the development of CT skills and programming in the schools. This chapter provides an in-depth analysis of policies of CT in the education of one particular country, namely Singapore. We review Singapore's approach to its implementation of CT education by first describing various initiatives in Singapore for Preschool, Primary and Secondary schools. Unlike several countries that have decided to implement computing as compulsory education, Singapore has taken a route of creating interest amongst children in computing in age-appropriate ways. Singapore's pragmatic approach of relying on an ecosystem is characterised by allowing schools the choice to opt-in, nurturing students' interest in computing, upskilling teachers in computing and a multi-agency approach.

Keywords Singapore programming initiatives · Computational thinking · Ecosystem in learning computing · Physical computing

P. Seow (✉) · C.-K. Looi · M.-L. How · L.-K. Wu
National Institute of Education, Singapore, Singapore
e-mail: peter.seow@nie.edu.sg

C.-K. Looi
e-mail: cheekit.looi@nie.edu.sg

M.-L. How
e-mail: mengleong.how@nie.edu.sg

L.-K. Wu
e-mail: longkai.wu@nie.edu.sg

B. Wadhwa
National University of Singapore, Singapore, Singapore
e-mail: bimlesh@nus.edu.sg

19.1 Introduction

Since Wing's (2006) argument on how computational concepts, methods and tools can develop thinking skills to transform how we work or solve problems, and with the emergence of computation-related fields such as Data Science and Artificial Intelligence in recent years, there has been a great interest from academia, industry and government in Computational Thinking (CT) and programming. Sites such as code.org, which is sponsored by industry giants like Google, provide free resources on learning programming to anyone who is interested. Though research in programming and computing education has been around for a few decades going back to the introduction of Logo in the 1970s, there is a renewed interest in learning programming and how it develops CT skills. National governments in addressing the shift from a knowledge/information economy to an economy driven by computation, are introducing educational policies that would prepare their citizens to be future ready. Reflecting 10 years after her seminal publication on CT, Wing (2017) said she never dreamt that Computer Science education, which was once available only at university level, would be introduced in K-12 at such a large scale today. Governments, educational authorities and schools are introducing Computer Science education at different levels of education. In countries such as the United Kingdom, Lithuania, Finland, Korea and Japan, initiatives and policies are made to introduce Computational Thinking skills and programming in the schools. This paper describes Singapore's effort in introducing the CT and programming in the education from preschool to secondary schools.

19.2 International Initiatives

England is one of the first countries to implement Computational Thinking in its K-12 curriculum (Bocconi, Chiocciariello, & Earp, 2018). In 2014, the curriculum was reformed to infuse CT into the curriculum, and it is organised in four key stages over the span of formal K-12 education in the UK. For each stage, students are expected to develop aspects of Computational Thinking skills progressively. For example, in Key Stage 1 (age 5–7 years old), students create and debug simple programmes, in Key Stage 2 (age 7–11 years old), they can design, write and debug programmes to accomplish specific goals, in Key Stage 3 (age 11–14 years old), students can design, use and evaluate computational abstractions that model the behaviour of problems and at Key Stage 4 (age 14–16), students develop and apply their analytical, problem-solving, design and computational thinking skills (DfE, 2017). The development of CT skills is integrated in the curriculum, paving the way for a reform in how subjects such as language, mathematics and science are taught.

In Europe, Finland has introduced algorithmic thinking and programming as a compulsory cross-curricular activity from grade 1. The development of a new National Core curriculum for primary and lower secondary schools in 2014 included

learning objectives that relate to aspects of CT and programming, as well as developing problem-solving skills in the context of real-world problems.

In Asia, countries like Japan and Korea are planning to make programming a compulsory component of education in schools. Japan recently announced plans to make computer programming compulsory for all primary school children by 2020, followed by middle school students in 2021 and high school students by 2022 (*Japan Times*, 2017). South Korea prepares students for its Creative Economy with strategies such as its Software Education initiative. The changes in the curriculum is focused on developing skills, CT and creative expression through programming, which will be implemented at all levels from primary to university education. The new programme will become mandatory in 2018 for the primary and lower secondary levels (APFC, 2017).

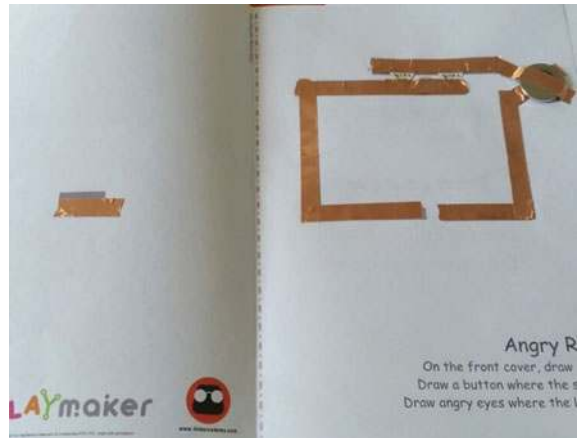
19.3 Computing Programmes in K-10

In 2014, Singapore launched the Smart Nation initiative, a nationwide effort to harness technology in the 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). One of the key enablers, to support the above initiative, is to develop the nation's computational capabilities. Programmes are implemented to introduce and develop CT skills and programming capabilities from preschool children to adults. We survey the landscape of K-10 CT and programming-related curricula in Singapore, which are implemented by various government organisations. We present these programmes, organised according to three groups: Preschool, Primary and Secondary.

19.3.1 Preschool

In Singapore, children aged from 3 to 6 years old attend preschools which are mostly privately run. The Infocomm Media Development Authority (IMDA) launched the Playmaker initiative with the aim of introducing Computational Thinking in the Kindergarten and Preschools in Singapore (IMDA, 2017). There are over 3000 preschools in Singapore. The initial phase involved piloting the programme in 160 preschools. IMDAs approach to introducing CT is to use electronic, robotic or programmable toys that would engage young children in play while developing CT skills such as algorithmic thinking. IMDA provided a set of the toys to pilot centres for use in the classroom by the teachers.

The toys selected by IMDA for playful exploration of technology are: (1) Beebot; (2) Circuit Stickers; and (3) Kibo. The Beebot is a toy with simple programmable steps to control its movement. Children can program the toy to move it along a path by logically sequencing the number of steps to move and control its direction.

Fig. 19.1 Circuit stickers

Playing Beebot can help young children to develop problem-solving skills and logical thinking as they plan and program the movement of the toy. Kibo, developed by researchers in Tuft University, allowed children to create a sequence of instructions by arranging Kibo wooden blocks. The blocks can be scanned in a sequence with the instructions passed to the robot to execute the steps. Circuit sticker is a toolkit comprised of peel-and-stick electronic components such as LEDs and conductive copper tapes. With this toolkit, young children can create interactive art and craft projects embedded with LED stickers and sensors that respond to the environment or external stimuli (see Fig. 19.1). Children can creatively ‘make’ while learning and applying basic electricity concepts.

Preschool teachers in Singapore do not use much technology or handle technology in the classroom as the emphasis is more on literacy development and play. As a result, they typically have apprehensions or concerns in using technology in their lessons. To address teachers’ lack of experience and concerns, IMDA organises teacher seminars and workshops for teachers to experience the use of the Beebot, Kibo and Circuit Stickers. The hands-on sessions are facilitated by the instructors to introduce teachers to the tech toys and work on simple projects. These experiences enable teachers to understand the potential learning opportunities for their students by first learning the technology for themselves. Hands-on sessions alleviate any potential fear of handling technology as they experience the use of the technology with the support from instructors.

In preparing to pilot the Playmaker program and address the concerns of preschool teachers, IMDA works with a local polytechnic which offers preschool training for teachers. At the Preschool Learning Academy, preschool lecturers and trainers, together with technologists work together to formulate the use of the various tech toys in the preschool classroom. The learning experiences were shared among the teachers. Such a collaboration created an opportunity to understand how the tools could be used in the classroom and it also built the capacity among the trainers to work with the teachers on how these tools can be used to develop the students’ learn-

ing potential. The academy was able to provide ongoing professional development to the current and new teachers.

Marina Bers and Amanda Sullivan were engaged by the IMDA to study the effectiveness of the KIBO program in the preschools. They studied the preschools' implementation of a curriculum called 'Dances from Around the World'. They found that the children were successful in mastering the foundational concepts of programming and that the teachers were successful in promoting a creative environment (Sullivan & Bers, 2017).

19.3.2 Primary Schools

To expose and enthuse primary school students in computational thinking, IMDA introduced its Code for Fun enrichment programme, which was piloted in 2014. As of March 2016, the programme has been implemented in 117 schools with about 34,000 participating students. The goals of the programme are to expose a large base of students to CT concepts and programming, develop a generation of workforce equipped with basic programming and CT skills. To scale-up this enrichment programme, IMDA invited technology training partners to propose 10 h programmes that would include programming activities using a visual programming language, such as Scratch, combined with a robotic kit such as the MoWay or microcontrollers such as the micro:bit. The programme aims to make students appreciate programming and develop CT skills in problem-solving and logical thinking. Schools interested in the Code for Fun programme can select from the list of vendors and apply for funding from IMDA to run the programme in the school. At present, IMDA provides 70% of the funding for each student with the rest of the funds provided by the school, on the condition that a certain number of students will be attending the programme. Teachers are also required to attend a training course about the programme conducted by the technology vendors. IMDA aims that the programme would be taught by the trained teachers in the future. Currently, each 10-h session is conducted by the technology trainers in the respective school lab. In each session, students are introduced to computing concepts such as the use of variables and conditional statements, and on how to use a visual programming tool such as Scratch. Students also use robotic tools such as the Lego WeDo kits, or the MoWay robot based on the proposal by the different training partners. Schools can choose different tools offered by the various trainers based on their students' interest and budget.

The Code for Fun enrichment and Playmaker programme are part of the Code@SG movement initiated by the government to teach CT and programming to students from an early age. Driven by the IMDA, the initiative is important for building Singapore's national capacity of a skilled workforce by creating interest in the Computational skills and promoting Infocomm as a career choice. A multi-pronged approach of working with different partners involves the development of enrichment programmes, school Infocomm clubs and competitions. Previously, students in the school Infocomm clubs were responsible for operating the school's audiovisual and computer

equipment during events. The clubs are now organised such that students now have more opportunities to apply their creativity with computers through programming and digital media. IMDA and the Information Technology Standards (ITSC) committee also organises an annual programming competition CodeXtreme for students, supported by educational institutions such as the universities, polytechnics, the Singapore Science Centre and technology industry partners such as RedHat and PayPal. Students are encouraged to participate in CodeXtreme in which Primary school students can work in teams to complete a given project with Scratch. Each team has an adult as a supervisor and mentor. Prior to the competition, the students must attend a workshop to equip themselves with the necessary skills for the challenge of the hackathon.

19.3.3 Secondary Schools

In 2017, the Ministry of Education introduced a new Computing subject which will be offered to students as an ‘O’ Level subject. It replaced the existing Computer Studies subject (MOE, 2017). Students taking the subject would learn to code in Python, a programming language which was previously taught only at ‘A’ Level Computing. In the new syllabus design, students will develop CT and programming skills to create solutions with technology to solve problems. In the old Computer Studies syllabus, students were learning to be users of technology such as using software applications and understanding aspects of technology.

The new Computing syllabus is built on the framework shown in Fig. 19.2: (1) Computer as a Science; (2) Computer as a Tool; and (3) Computer in Society.

The dimension of Computer as a Science is comprised of the core components of Computational and Systems Thinking. Students will develop and apply CT skills such as abstraction and algorithmic thinking to solve problems and develop solutions through programming. Using both CT skills and systems thinking, students are required to work on a project of their own interest. This is, however, a non-assessment component of the programme. It is to encourage students to take more ownership by identifying a problem that they have an interest and develop ideas to solve the problem using programming tools. The purpose of a non-assessed project work is to encourage the students to be more creative in designing solutions without the pressure of assessment. In the dimension of Computer as a Tool, students are exposed to the use of hardware, technology, and devices that are used in the everyday aspects of life at work and play. They learn about computer applications that are used for productivity, communications and creative tools for completing specific tasks such as video editing or creating websites. In Computer in Society, students learn about issues in using computers such as intellectual property, data privacy, Internet security and the computer addiction. This dimension includes a component on twenty-first century competencies to prepare students to be future-ready workers in the use of technology for self-directed learning, working in collaboration with others and fostering creativity.



Fig. 19.2 Computing education framework

Currently, 22 out of about 150 secondary schools (15%) are offering Computing as an O level subject. One of the reasons for the low number of schools is the lack of teachers who can teach computing. There are relatively few teachers who have Computing or Computer Science background who can teach programming. Teachers who are interested in teaching Computing and programming attend a year-long conversion course taught by Computer Science faculty from a university. They were given time by their schools to attend the course during school hours. In the course, they upgraded their Computer Science content knowledge such as data structures and programming in Python. The goal of the course is to prepare and equip teachers with the content and technical knowledge to teach computing. In addition to preparing teachers for the new Computing curriculum, Ministry of Education’s Curriculum Planning and Development Division (CPDD) organised workshops for teachers to understand the aspects of the syllabus. Teachers are introduced to different pedagogies for teaching computing such as unplugged approaches and paired programming. In the workshops, teachers also experienced the use of the tools for teaching such

as the Raspberry Pi. The workshop also served as a platform for teachers to raise their concerns about teaching the subject, for example about how the project work for students can be organised and implemented in the school.

In most schools, the typical number of students offered Computing ranges from 15 to 30 students. Computing is offered as an elective subject for the O levels and student can opt-in for the subject at the end of Secondary 2. A student is typically offered 7–8 ‘O’ Level subjects in their Secondary 3 year. Subjects like Additional Mathematics and the Sciences are more favoured by students as these subjects are better considered for admission for pre-university junior colleges and polytechnics. Hence, students who are initially interested in taking up Computing may instead choose to take Additional Mathematics, because of the limit in the number of subjects they are offered and for pragmatic reasons for entry to pre-university courses.

19.4 Singapore’s Approach in Computing

Unlike countries like Finland, England and Korea, Singapore is not including Computing or CT as compulsory education. Instead, Singapore’s approach is to provide opportunities for students to develop their interests in programming and computing skills through touchpoint activities at various ages, as shown in Fig. 19.3. Computing and CT skills, which are introduced to the children are age-appropriate and can engage them in learning. Children can then progressively develop interest and skills, leading them to select Computing as a subject in the ‘O’ levels. The following sections describe the characteristics of the approach.

19.4.1 Opt-in by Schools

The decision to teach Computational Thinking skills and computing is decided by individual schools. An opt-in model respects the choice of each school to select programmes according to the needs of their students and the readiness of the teachers. Schools are encouraged to develop a niche in skills or sports, creating a diverse learning landscape to meet various educational needs and interest of students in Singapore. School-based programmes are planned by the school and teachers that would build students’ interest and skills in identified areas like Computing. As teachers play a pivotal role in implementing the programmes, there must be buy-in from the teachers who can see the importance of the programmes for the students. For the schools to opt-into adopt computing, there must be teachers within the school to be ready to learn, experiment, and implement the programme. One such example of opt-in is Bukit View Secondary School, Singapore. The school has formulated a programme that enables all students to learn computational thinking. The school addressed the issue of a packed curriculum by structuring time for learning computing in the time table and integrating with school subjects. Training partners are engaged to work

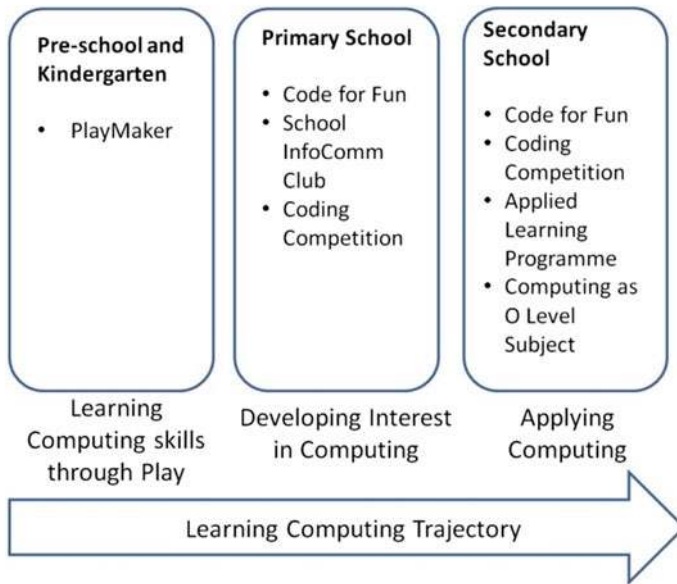


Fig. 19.3 Learning computing in Singapore

with teachers in designing and integrating computing with the subjects to address the lack of teachers' experience. Teachers' capacity is continuously developed as they gain competence in computing. The school iteratively improve the programme for sustainability and richer learning experience for the students (Hameed et al., 2018)

19.4.2 *Nurturing Interest in Computing*

Instead of making computing part of compulsory education, an approach is to nurture interest at an early age. Preschool children are developing problem-solving and logical thinking skills through play. Toys like the Beebot and Circuit Stickers are age-appropriate. Young children are engaged through play in their lessons while developing the computational thinking skills. In primary and secondary schools, students are introduced to visual programming tools like Scratch and tangible computing tools like the MoWay robots and Lego WeDo. Lessons are designed to have fun and engagement while learning programming and developing logical thinking skills. As students progress up to the O Levels, they can select Computing as a subject based on their interest and choice. Starting from preschool, a pipeline is created for students to develop interest and computational thinking skills; enabling them to choose Computing rather than making its learning compulsory.

19.4.3 Upskilling Teachers in Teaching Computing

To prepare teachers to develop logical thinking, algorithmic thinking, problem-solving and programming skills, professional development and support must be given. Professional development should be appropriate to the needs of the teachers; to prepare them to think about how they can teach their students. For the preschool teachers, a form of learning is for them to experience playing with the toys and in turn, understand how their students would learn from playing. Support from IMDA is given to the teachers to help them design and implement the lessons in the classrooms. In secondary schools, computing teachers undergo an intensive computing course learning in computer science concepts and programming. Most of these teachers are non-computer science graduates but they volunteer for the conversion course based on their own interests. Even after the course, these teachers hold regular meetups to continue improving their knowledge in teaching computing.

In the National Institute of Education, researchers are working with teachers to develop and design pedagogies for Computation Thinking and Computing Education. One such effort is the development of unplugged approaches for teaching computational thinking in the schools (Looi, How, Wu, Seow, & Liu, 2018). Researchers conduct workshops for computing teachers to experience various unplugged activities such as sorting with a balance scale, binary cards and deadlock avoidance (Fig. 19.4). Through the workshop experiences, teachers appreciate the role of kinaesthetic activities and concrete materials play in helping students to understand some of the key concepts in Computing. Appropriate unplugged computing activities that can be mapped to the computing curriculum are identified and designed. These unplugged computing activities are enacted in the classroom in collaboration with the teachers. Data regarding student learning outcomes from the unplugged activities are collected, analysed, and shared with the teachers. Using a collaborative design-based research approach, the researchers aim to bridge the gap of research and practice that can improve teaching practices in computing.

19.4.4 Multi-agency and Many Hands Approach

The task of building CT and Computing skills requires the combined effort of multiple agencies working together. They include the government agencies like Infocomm Media Development Authority (IMDA), Ministry of Education and the Ministry of Social and Family Development, education centres like the Singapore Science Centre, universities and educational training providers. These agencies have been working together, sometimes also independently, to organise opportunities for students to learn computational thinking skills by providing them with varied experiences. These agencies can pool resources such as funding and support for initiating, implementing and sustaining the programmes.



Fig. 19.4 Unplugged computational thinking activity with pre-service teachers

IMDA started the Digital Maker programme to develop a peoples' passion to create with technology and grow a community of digital creators. In 2017, IMDA introduced the micro:bit board, a pocket-sized programmable device, to the community including schools, teachers and students. Programmes and workshops are organised with training partners to explore the possibilities of digital making with the micro:bit boards. In schools, students can use the micro:bit boards to develop various real-world physical applications such as creating games, controlling a car, and measuring the height of the flag pole. In workshops for teachers, they learn the basics of using the micro:bit and how it can be used as a learning tool in the classroom. If teachers see value in the use of the boards and would like to receive a set of micro:bit boards for their school, they can submit a lesson plan to IMDA to provide details about how they plan to integrate the micro:bit boards in their lessons. Schools can engage training partners to run the additional programmes or train more teachers in the use of the micro:bit boards. The above example shows how different partners can work together in an ecosystem to teaching Computing in the schools. In the next section, we will elaborate on this ecosystem in computing education.

19.5 Towards an Ecosystem of Learning Computing

A learning ecosystem is defined as a network of relationships among learning agents, learners, resources, and assets in a specific social, economic and geographic context (Prince, Saveri, & Swanson, 2015). Learning for students is not limited by the boundaries of the classroom and lessons taught by teachers but facilitated through a network

of resources and agents to develop learning pathways based on the interest of learners. Students have opportunities to informal and formal resources in the learning ecosystem (Basham, Israel, & Maynard, 2010). They can participate in enrichment programmes, special-interest clubs and after-school activities; have access to a myriad of digital based media such as videos, social media and Internet. In the learning ecosystem, each agent plays an important role in the ecosystem in initiating and sustaining the learning interest of learners. Compared to the traditional classroom where tightly bound relationships and resources are the nexus; the flow to deliver instruction, develop curriculum, perform assessment, teaching, and learning can now be available, through and enhanced by a vibrant learning ecosystem of resources for students.

Bell, Curzon, Cutts, Dagiene, and Haberman (2011) explain that informal outreach programmes which downplay syntax-based software programming as a prerequisite skill for engaging with Computer Science ideas can effectively make CT concepts accessible to the students in short bursts without formal curriculum support. These outreach programmes can operate outside, or in conjunction with the formal education system to expose students to computer science concepts, so that they can make informed decisions about their career paths. Kafai and Burke (2013) observe that developments in K-12 computer software programming education can be characterised by a 'social turn', a shift in the field in which learning to code has shifted from being a predominantly individualistic and tool-oriented approach to now one that is decidedly sociologically and culturally grounded in the creation and sharing of digital media. The three dimensions of this social turn are: (1) from writing code to creating applications, (2) from composing 'from scratch' to remixing the work of others and (3) from designing tools to facilitating communities. These three shifts illustrate how the development of artefacts, tools and communities of programming could catalyse the move from computational thinking to computational participation, and hence broaden the students' participation in computing. One of the most active proponents of this social turn is the Maker Movement (Martin, 2015), a community of hobbyists, tinkerers, engineers, hackers, and artists who creatively design and build projects for both playful and useful ends. There is a growing interest among educators in bringing 'making' into K-12 education to enhance opportunities to utilise CT concepts to engage in the practices of engineering, specifically, and STEM more broadly. CT is considered the best practice for teaching computing and more broadly to solve problems and design systems; however, as computing extends beyond the desktop, Rode et al. (2015) argue we could transition from CT to computational making as an educational framework via the Maker Movement. Besides using micro:bit, many people in the Maker Movement also use the LilyPad Arduino, a fabric-based construction kit that enables novices to design and build their own soft wearables and other textile artefacts. (Buechley, Eisenberg, Catchen, & Crockett, 2008; Kafai et al., 2013; Kafai, Lee, Searle, & Fields, 2014).

In Singapore, students can choose to participate in the Learning Ecosystem for Computing, which is comprised of informal education programmes such as Code for Fun, computing-related Core Curricular Activities (CCA) such as the Computer and Robotics club, and in computer-based Applied Learning Programmes (ALP). Nevertheless, there is a dearth in the extant literature about how these informal learning educational programmes could help to sustain the students’ interests in the development of CT after they have progressed through them. Low (2014) notes that while ALP holds promise in achieving authentic achievements as a long term and sustainable form of learning that can positively influence student learning, equip students with twenty-first century competencies and ensure positive future outcomes, there is a need for greater scholarship discourse in areas of authentic assessment and authentic achievement and their roles in the enactment of ALP.

The various programmes provide varied experiences for the students to develop their own interests and apply computing to solving problems. The programmes are facilitated, developed and implemented by stakeholders in the ecosystem as shown in Fig. 19.5. The agents in the ecosystem directly involved with students include teachers, school leaders and parents. External agents include the government agencies such as IMDA, Ministry of Education, the Singapore Science Centre and the computing trainers. The range of programmes for Computing includes formal Computing education in the O and A level computing; informal programmes in schools such as Code for Fun, Play/Digital Maker, Computing-based CCAs and ALP; and the outside-school enrichment programmes such as Computing Educational Centres or the Singapore Science Centre (Fig. 19.5).

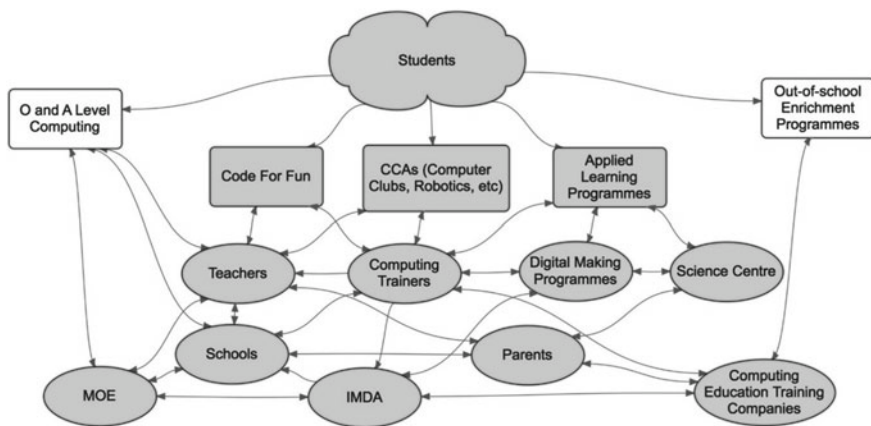


Fig. 19.5 Ecosystem for learning computing in Singapore

Currently, computing in formal education is offered at 'O' Level and 'A' Level curriculum. Pixel labs under Code@SG movement (Pixel Labs, 2017) aims to see programming and CT as Singapore's national capability. Accordingly, programming and CT are being considered to teach from an early age to students. Children with special needs and the underprivileged are also encouraged to cultivate their interest in programming, programming and technology development. The Infocomm Clubs (Infocomm Club, 2017), a co-curricular activity programme for school children not only excites students about Infocomm in a fun and meaningful way but also cultivates leadership and entrepreneurship capabilities along with providing opportunities for project work and community service projects at an early age. Many initiatives (Code in Community, 2017) offers free programming lessons to underprivileged children in Singapore. Thus, the main approach is to enthuse a broad base of students in computing and expose them to possibilities of technology through enrichment programmes and co-curricular activities. Learning to code is part of the ALP in 41 secondary schools. In addition, MOE also partners IMDA to provide enrichment programmes like the 'Code for Fun' and 'Lab on Wheels', which have been well received by schools. There are secondary schools (33 schools in 2016) with Infocomm clubs, which tap on the support of IMDA to provide learning in areas involving programming such as app development and robotics.

In the National Institute of Education, efforts have been made to bring together researchers, teachers, computing education trainers, and students together to share ideas and practices in the learning of Computing and Computational Thinking. In 2017, a half-day symposium and workshop was organised within a major international education conference with representatives from Infocomm Media Development Authority (IMDA), National Institute of Education (NIE), National University of Singapore (NUS), school teachers and students presented national-level initiatives on Computing Education, research findings, teaching and learning experiences in Computing. A workshop provided attendees to participate in various Computing learning activities such as unplugged activities, board game playing and physical computing with micro:bit and Arduino (Fig. 19.6). We plan to continually organise more of these events to draw participants in the ecosystem together in the sharing, learning and building of a vibrant community for Computing Education in Singapore.

19.6 Summary

Singapore has taken a pragmatic approach in the implementation of CT skills and computer science education. Taking such an approach provides children with the opportunities to generate interest in learning computing. Starting at an early age, children are exposed to developing CT skills through age-appropriate ways of playing. In primary schools, students learn through fun and are given opportunities to extend their interest in programming through clubs and programming competition. At the secondary school, students can choose whether they wish to pursue Computing as a subject. Schools can opt-into offer programmes based on the students'

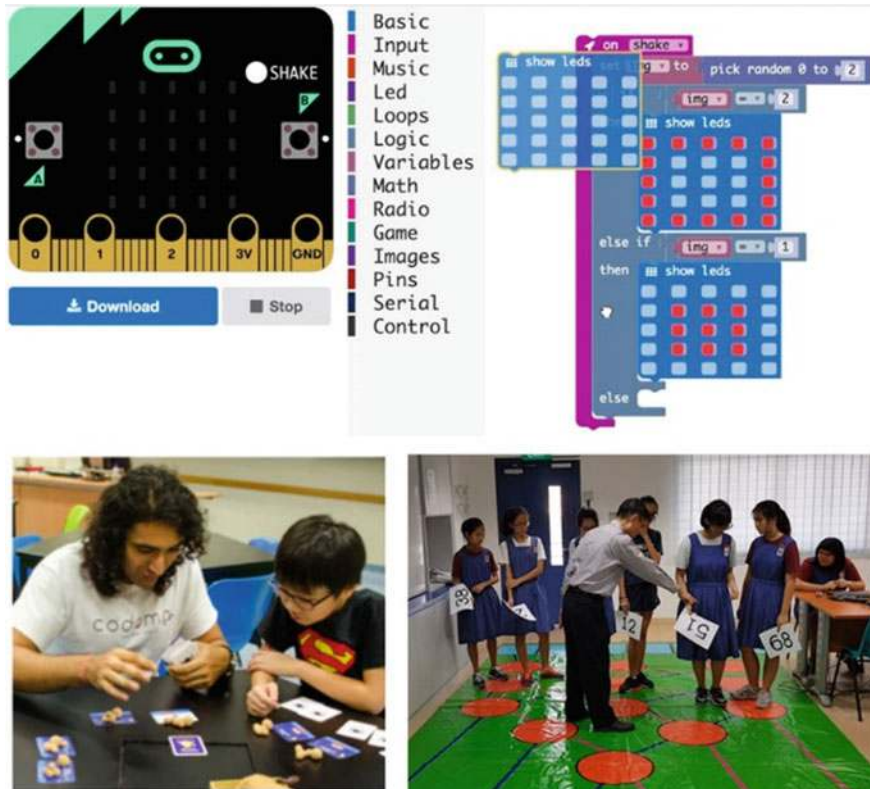


Fig. 19.6 Workshop with computing learning activities such as physical computing with micro:bit, card games and unplugged activities to learn computing concepts

needs, schools' niche programmes and readiness of the teachers to teach computing. Teachers who are keen can choose to extend their capacity to teach computing. Singapore as a nation can harness various agencies to work together in providing a variety of learning experiences for children to be engaged in computing learning. Thus, Singapore's approach to CT education is promising but different from various countries that have made CT part of compulsory education.

References

- APFC. (2017). *Preparing students for South Korea's creative economy: The successes and challenges of educational reform*. Retrieved February 13, 2017, from <http://www.asiapacific.ca/research-report/preparing-students-south-koreas-creative-economy-successes>.
- Basham, J. D., Israel, M., & Maynard, K. (2010). An ecological model of STEM education: Operationalizing STEM for all. *Journal of Special Education Technology*, 25(3), 9–19.

- Bell, T., Curzon, P., Cutts, Q., Dagiene, V., & Haberman, B. (2011). Overcoming obstacles to CS education by using non-programming outreach programmes. In *Informatics in schools: Contributing to 21st century education* (pp. 71–81). Springer. <https://doi.org/10.1007/978-3-642-24722-4>.
- Bocconi, S., Chiocciariello, A., & Earp, J. (2018). *The Nordic approach to introducing computational thinking and programming in compulsory education*. Report prepared for the Nordic@BETT2018 Steering Group. <https://doi.org/10.17471/54007>.
- Buechley, L., Eisenberg, M., Catchen, J., & Crockett, A. (2008). The LilyPad Arduino. In *Proceeding of the Twenty-Sixth Annual CHI Conference on Human Factors in Computing Systems—CHI '08* (p. 423). <https://doi.org/10.1145/1357054.1357123>.
- Code in Community. (2017). Google's free coding classes are proving to be a major success—Sees first 500 S'pore kids graduate. Retrieved May 22, 2017, from <https://vulcanpost.com/611220/code-community-google/>.
- DfE. (2017). *National curriculum in England: Computing programmes of study*. Retrieved February 13, 2017, from <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>.
- Hameed, S., Low, C.-W., Lee, P.-T., Mohamed, N., Ng, W.-B., Seow, P., & Wadhwa, B. (2018). *A school-wide approach to infusing Coding in the Curriculum* (pp. 33–36). Paper presented at the Proceedings of 2nd Computational Thinking Education Conference 2018, Hong Kong SAR, The Education University of Hong Kong.
- IMDA. (2017). *PlayMaker changing the game*. Retrieved February 13, 2017, from <https://www.imda.gov.sg/infocomm-and-media-news/buzz-central/2015/10/playmaker-changing-the-game>.
- InfoComm Club. (2017). *Infocomm Clubs programme*. Retrieved February 13, 2017, from <https://www.imda.gov.sg/imtalent/student-programmes/infocomm-clubs>.
- Japan Times*. (2017). *Computer programming seen as key to Japan's place in 'fourth industrial revolution'*. Retrieved February 13, 2017, from http://www.japantimes.co.jp/news/2016/06/10/business/tech/computer-programming-industry-seen-key-japans-place-fourth-industrial-revolution/#.WKG2P_I97b0.
- Kafai, Y. B., & Burke, Q. (2013). The social turn in K-12 programming. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education—SIGCSE '13* (p. 603). <https://doi.org/10.1145/2445196.2445373>.
- Kafai, Y. B., Searle, K., Kaplan, E., Fields, D., Lee, E., & Lui, D. (2013). Cupcake cushions, scooby doo shirts, and soft boomboxes: e-textiles in high school to promote computational concepts, practices, and perceptions. In *Proceedings of the 44th ACM technical symposium on Computer Science Education* (pp. 311–316). ACM.
- Kafai, Y. B., Lee, E., Searle, K., Fields, D., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE)*, 14(1), 1.
- Looi, C. K., How, M. L., Wu, L., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Journal of Computer Science Education (JCSE)*. <https://doi.org/10.1080/08993408.2018.1533297>.
- Low, J. M. (2014). Applied learning programme (ALP): A possible enactment of achieving authentic learning in Singapore schools. In *40th Annual Conference (2014)*. Singapore: International Association for Educational Assessment. Retrieved from http://www.iaea.info/documents/paper_371f25129.pdf.
- Martin, L. (2015). The promise of the maker movement for education. *Journal of Pre-College Engineering Education Research (J-PEER)*, 5(1). <https://doi.org/10.7771/2157-9288.1099>.
- MOE. (2017). *O level computing teaching and learning syllabus*. Retrieved February 13, 2017, from <https://www.moe.gov.sg/docs/default-source/document/education/syllabuses/sciences/files/o-level-computing-teaching-and-learning-syllabus.pdf>.
- Pixel Labs. (2017). *CODE@SG movement—Developing computational thinking as a national capability*. Retrieved October 10, 2017, from <https://www.imda.gov.sg/industry-development/>

[programmes-and-grants/startups/programmes/code-sg-movement-developing-computational-thinking-as-a-national-capability](#).

- Prince, K., Saveri, A., & Swanson, J. (2015). *Cultivating interconnections for vibrant and equitable learning systems*. KnowledgeWorks: Cincinnati, OH, USA.
- Rode, J. A., Weibert, A., Marshall, A., Aal, K., von Rekowski, T., El Mimouni, H., & Booker, J. (2015, September). From computational thinking to computational making. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 239–250). ACM.
- Smart Nation. (2017). *Why smart nation*. Retrieved February 13, 2017, from <https://www.smartnation.sg/about-smart-nation>.
- Sullivan, A., & Bers, M. U. (2017). Dancing robots: Integrating art, music, and robotics in Singapore’s early childhood centers. *International Journal of Technology and Design Education*, 1–22.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2017). *Computational thinking, 10 years*. Retrieved December 13, 2017, from <https://www.microsoft.com/en-us/research/blog/computational-thinking-10-years-later/>.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

