CODING EDUCATION FOR KIDS: WHAT TO LEARN? HOW TO PREPARE TEACHERS?

Wing Shui Ng
The Education University of Hong Kong
Hong Kong, China

Abstract

In the area of science, technology, engineering and mathematics (STEM) education, coding has been highlighted as a core competence. Young children are required to start to learn coding. However, coding is not an easy skill, and early childhood educators usually lack relevant training. This paper suggests that the learning of logical concepts and sequential program are appropriate for young children. This paper also reports a case study of teacher training. Preliminary results suggest that early childhood educators are able to design appropriate learning activities of coding education for young children after receiving training.

Introduction

The new initiative of science, technology, engineering and mathematics (STEM) education has been advocated in recent years, and coding skill is particularly highlighted as an essential competence. It has been suggested that children should start to learn coding at age five (Department for Education, 2014; Smith, 2016). The importance of coding is considered comparable with reading, writing and arithmetic. However, to implement coding education in kindergarten is not a straightforward issue. As coding in computer science involves abstract concepts and complicated skills, and children are still in an early developmental stage, it is critical to explore appropriate learning contents that are suitable for young children. On the other hand, training on coding education is seldom included in a teacher training programme. Teachers, therefore, may have difficulties to design learning activities and deliver lectures on coding education for children (Bers, Seddighin, & Sullivan, 2013; Manches & Plowman, 2017). In consideration of the new initiative of promoting coding education to children, this paper attempts to discuss the role of ICT education, and particularly coding skill, in STEM education. The researcher then attempts to explore the learning contents of coding education that are appropriate for children at age five. Moreover, this paper reports a case study of a training program to prepare teachers for implementing coding education in an early childhood setting.

ICT Education in STEM Education

In the past few years, a lot of countries have proposed policies to promote STEM education. In the United States, the Department of Education (2017) clearly highlighted the aim to maintain global leadership by strengthening the learning and teaching of STEM contents. Government officials stressed that in a complex world, youth are required to know what can be done on the basis of obtained knowledge so as to achieve success. Youth are particularly expected to obtain knowledge and skills in STEM education to solve problem and process information. In response to the educational need, strategic policies for STEM education have been proposed in the United States. In Singapore, STEM education has been integrated into the Applied Learning Programme (Ministry of Education, 2017). The programme aims to provide learning opportunities for students to apply their knowledge and skills in science, mathematics and technology to solve real world problems. Learning contents include scientific inquiry and literacy, reasoning and problem solving, design thinking, computational thinking, data analysis and the use of technology. In Hong Kong, the Curriculum Development Council (2015) has published an official document to promote STEM education. Moreover, the Chief Executive stated in The 2017 Policy Address (HKSAR, 2017) that the Education Bureau planned to provide additional resources to schools to facilitate the implementation of school-based programmes related to STEM education. Also in England, learning contents related to STEM have been integrated in different key stages of the national curriculum (Department for Education, 2014).

Within the domain of STEM education, information and communication technologies (ICT) education has been recognized as an important area of learning (Manches & Plowman, 2017). To explore the relationship between ICT and STEM education, the author identified four key roles of ICT education to support the broader STEM education (Figure 1).

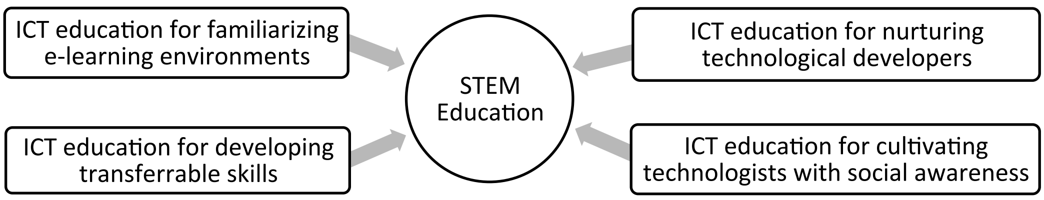

Figure 1. Roles of ICT education in STEM education.

The first role of ICT education is to help develop transferrable skills. It includes two important areas of transferrable skills, namely information literacy and computational thinking. With the advancement of technologies and rapid development of the Internet, information explodes in an exponential rate, and it boosted the formation of today's knowledge society. Students are, therefore, required to develop necessary literacy to deal with a huge volume of information in their learning. Information literacy refers to "the abilities to recognize when information is needed and to locate, evaluate, effectively use, and communicate information in its various formats" (State University of New York, 1997, para. 5). An information literate is expected with competence to manipulate a wide range of information processing tools. Students receiving STEM education are usually engaged in a process of inquiry by searching, comprehending, organizing, synthesizing and evaluating information for knowledge construction. For example in Oldknow's (2009) study, a lot of information processing tools, such as digital image and video editing software, 3D modeling software and data logger software, were effectively used in a series of ICT-based learning activities for learning mathematical concepts. In another study conducted by Awad and Barak (2016), simulation and sound editing software were adopted in STEM lessons to learn the concepts of sound waves. In addition, the WebQuest model that analyzes information obtained from the web for inquiry learning has been commonly used in STEM classrooms (Osman, 2014). Information literacy is, therefore, considered as a critical transferrable skill in STEM education. Another important transferrable skill that can be developed by ICT education is computational thinking. According to Wing (2006), "Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science" (p. 33). Barr and Stephenson (2011) further expressed the idea with the following elaboration:

> Computational thinking is an approach of solving problems in a way that can be implemented with a computer. Students … use a set of concepts, such as abstraction, recursion, and iteration, to process and analyze data, and to create real and virtual artifacts. Computational thinking is a problem solving methodology that can be automated and transferred and applied across subjects. (p. 51)

In recent years, researchers have started to put forth efforts to design learning activities in school for developing students' computational thinking skill. For example, Lee and her colleagues (2011) designed learning activities for training computation thinking in the domains of modeling and simulation, robotics, and game design and development in STEM education. Computational thinking has been advocated as an essential skill for everyone in a digital age (Barr, Harrison, & Conery, 2011).

Another role of ICT education in STEM education is to enable learners to become familiar with e-learning environments. It includes the knowledge and skills to recognize the functionalities afforded by various e-learning platforms, to adapt or construct e-learning platforms, and to appropriately adopt them to support learning. For example, a wiki is a simple and easy-to-use web-based platform for cooperative work. Learners are encouraged to develop a wiki site to construct knowledge collaboratively (Ebersbach, Glaser, & Heigl, 2006). Wiki has been used as a learning environment to engage students in learning STEM related contents (Ng, 2016). In addition, a 3D immersive virtual environment was adopted to engage students in STEM lessons in the study by Roberts and Matzen (2013). A content management system, such as Moodle, is another example of an e-learning platform to support online learning. With a view to facilitate students' learning in STEM education, ICT education should equip students with knowledge on the functionalities of e-learning platforms and enable them to make good use of them.

The third role of ICT education in STEM education is for nurturing technological developers. It involves computer programming skill, logical reasoning skill and knowledge of software development. An important application of the computer programming competence is the development of mobile apps. Mobile devices have become very popular due to their strong computing capability, small size, Internet connectivity and multiple functionalities. People have paid more attention to the development of mobile apps due to their possibilities of gaining benefits (Hsu, Rice, & Dawley, 2012). On the other hand, computer programming is often associated with robotics. A series of instructions in a computer program is used to control the behaviors of a robot for a specific purpose. Recent studies can be found to integrate computer programming and robotics elements in STEM lessons to equip students with technological knowledge and skills (Park, Kim, Oh, Jang, & Lim, 2015; Sullivan & Bers, 2016). In that connection, ICT education paves the way for developing future technological developers.

The last important role of ICT education in STEM education is to cultivate technologists with social awareness. Although knowledge and skills are important, it is also critical to cultivate appropriate values and attitudes in STEM education (Rao, 2014). To address the issue of social awareness, ICT education should emphasize the appropriate use of technology, including proper netiquette in online activities, the respect of intellectual property, responsible use of technology, the awareness of equity and different cultures, and a good sense on the impact of technology on health and environment.

Coding in STEM Education

Computer programming and *coding* are two associated terms in computer science. According to Manches and Plowman (2015) coding refers to “the specific skills of inputting instructions using a particular language” while programming refers to “the wider design and implementation process of using code to solving particular problems” (p. 3). Coding can be regarded as a set of fundamental skills for computer programming. In response to the need of STEM education, the previous President of the United States, Barack Obama, initiated the Computer Science for All policy in 2016 (Smith, 2016). This policy aimed to educate students to be technological developers and active citizens in the technology-driven world. Students are required to learn computer science from kindergarten to high school. Particularly, the importance of coding education in the digital economy was highlighted. In England, a new curriculum has been developed to mandate the inclusion of computer science learning contents for students at age five. The importance of coding is considered comparable with reading, writing and arithmetic (Department for Education, 2014). Therefore, it is generally believed that students can benefit from the learning of coding. However, the implementation of coding education for children at age five is not a straightforward issue. Efforts are required to tackle some challenges in the implementation.

Challenge on the Difficulties with Coding

Coding is not an easy skill to master. Among different aspects of difficulties, the hurdle coming from the linguistic intricacies of computer programming language has been addressed by many researchers (Gomes & Mendes, 2007). As mentioned by Gomes and Mendes, computer programming languages were developed for professional use with many complex syntactic details to memorize. The languages used for programming are hardly suitable for novices. In addition, learners are required to have good logical reasoning skills to deal with logical operations and control flow in a computer program. In this regard, many researchers have proposed methodologies and tools to help students master the difficulties on learning computer programming (Gomes & Mendes, 2007; Robins, Rountree, & Rountree, 2003,). Therefore, if children at age five are required to learn coding, it is critical to identify the learning elements and tools in coding education that are appropriate for their learning needs, and suitable for their cognitive development.

Challenge on Teachers' Competence for Coding Education

Another challenge refers to teachers' competence in the implementation of coding education for kids (Bers et al., 2013; Manches & Plowman). Due to different phases of technology development, early childhood educators are unlikely to have experiences of computing and particularly coding education from their own education or career (Manches & Plowman, 2017). They usually lack knowledge about technology and engineering, and they also do not have sufficient pedagogical knowledge to bring technology into the classrooms (Bers et al., 2013). As a consequence, they are not confident in their teaching, and it is hard for them to deliver high-quality STEM lessons.

Methodology

In view of the challenges, the author designed a teacher training program with a series of workshops on coding education for pre-service teachers in an early childhood educational setting. The training program was implemented in the 2015 – 2016 academic year in Hong Kong. This study aimed to investigate the effectiveness of the teacher training program. Details of the implementation and preliminary results of the case study are elaborated in the following sections.

Learning Elements in Coding Education for Kids

The first step of designing the teacher training program on coding education for kids was to identify the most appropriate learning content contents. To achieve the purpose, it is necessary to understand the cognitive development of children. According to the cognitive-development theory suggested by Piaget (1971), children aged five are at the preoperational stage. In this stage, a child begins to understand the world using mental representations with words and images. This kind of symbolic thinking goes beyond the connection of sensory information with physical action. Dewey (1938) and Piaget (1947/1960) believed that children develop knowledge through active participation in their learning based on their own experiences. Piaget (1971) particularly highlighted that the thinking of a child at the preoperational stage is characterized by the lack of logic. Although the results of some studies (Au, Sidle, & Rollins, 1993; Gelman, 1972; Rosen & Rozin, 1993) showed that children at preoperational stage could demonstrate satisfactory logical concept and reasoning ability, children are usually unable to produce accurate logical reasoning when faced with unfamiliar topics, too much information, or contradictory facts that they cannot reconcile (Berk & Meyers, 2016).

As reflected in the literature, the logical concept and reasoning skills of children at five are weak and still to be developed. Since logical concepts are critical in computer programming, the learning of fundamental logical concepts, such as logical AND, OR, NOT, and reasoning skills, such as IF, THEN deductive reasoning, are considered appropriate for children of age five. Additionally, almost all computer programming textbooks help learners start to learn with a simple program of sequential execution of statements (see for example, Deitel and Deitel, 2015). It suggests that a computer program with instructions in a direct sequence is the simplest structure and the learning of sequential programming is appropriate for children.

Teacher Training Program

Since children of age five are attending kindergarten in Hong Kong, a group of pre-service teachers taking the higher diploma of early childhood education programme in The Education University of Hong Kong were invited to participate in this study. Since this project was considered as a new initiative, the case study research method was adopted as it enables a group to be "studied extensively and varied data are collected and used to formulate interpretations applicable to the specific case or to provide useful generalizations" (Fraenkel, Wallen, & Hyun, 2012, pp.13-14). It also "aims to understand the case in depth, and in its natural setting, recognizing its complexity and its context" (Punch, 2009, p.119). A total of 10 pre-service early childhood teachers, 5 from year 1 and 5 from year 2, participated in this study. In addition, an IT team comprised of 2 year 4 students taking the bachelor of information and computer technology programme was formed to provide training on coding education. The researcher with strong background on IT in education and early childhood education served the role of overall supervisor of this project.

The training program consisted of three workshops.

Workshop 1. In the first workshop, to strengthen basic logical concepts and to master elementary coding skills were the main objectives. The design of the workshop was underpinned by the theory of the zone of proximal development (ZPD) and scaffolding suggested by Vygotsky (1978). Vygotsky recognized that the development process lags behind the learning process, and this results in the ZPD, which is critical for learning. In this concept, an essential feature of learning is to identify the ZPD and equip learners with the capacity to proceed to this zone by interacting with more knowledgeable others. In the workshop, the researcher delivered a lecture on some basic logical concepts. Although the participants would have developed basic logical concepts in their life experience, the lecture aimed to provide formal training to strengthen their logical concepts and reasoning skills. Participants then received a training session on elementary coding skills. Hands-on practices were integrated in the training workshop using technological tools for the purpose of scaffolding.

Workshop 2. In the second workshop, the researcher introduced some learning resources that are suitable for coding education for children. The researcher together with IT team members then provided a demonstration on how to apply related resources to design learning activities on coding education. As computer programming language is highly complicated, and not appropriate for the cognitive developmental stage of the children, a simple robot, Bee-Bot (Figure 2), was chosen as a learning tool to develop the children's ability to compose sequential instructions. A Bee-Bot is an easy-to-operate robot that can be moved on a mat based on inputted instructions. It has been adopted as a learning tool to develop programming skills in early childhood education (Eck et al., 2013; Janka, 2008). Since children learn better if learning activities are related to their life experience, the researcher intentionally designed a mat for the Bee-Bot with daily life objects (Figure 3). In the design of learning activities, children are required to answer logical questions and instruct the Bee-Bot to move to a specific location. An example of a logical question was "if today is a sunny day, please instruct the little bee to meet the sun." The overall design of learning activities served as a scaffolding example to facilitate teacher training. After this second workshop, the pre-service teachers were asked to design learning activities on coding education for kids.

Figure 2. A Bee-Bot robot movement.

Figure 3. A mat for Bee-Bots.

Workshop 3. In the third workshop, the pre-service teachers carried out a micro-teaching. The purpose was to engage participants to work together and to promote collaborative learning between pre-service teachers who have similar levels of conceptual understanding (Fernandez, Wegerif, Mercer, & Drummond, 2001). During the preparation of the learning activities and the micro-teaching, participants were encouraged to discuss with each other, to exchange ideas and to learn collaboratively. The design of the learning activities was evaluated by the researcher to serve as evidence to reflect on the effectiveness of the training program.

Results

In the micro-teaching, the participants designed a 3x3 square mat for the Bee-Bot with scenarios related to children's life experience (Figure 4), and they also designed a card, "Timetable of Little Bee," to facilitate the learning activities (Figure 5). In the design, the early childhood educator would first declare the day of week in the scenario and request a child to evaluate the statement like "If it is Tuesday, the little bee should go to the hospital and the market." The child would consider whether the day in the scenario is Tuesday. If it is Tuesday, the child should instruct the little bee to move to both locations consecutively. If it is not Tuesday, no action is required. If the child successfully completes the task, it suggests that the child understands the logical concept and is able to construct a series of instructions for developing fundamental coding skill.

Football Court	Library	Garden
Post Office	Bakery	Police Station
School	Supermarket	Hospital

Figure 4. A mat for Bee-Bot' movement designed by the participants.

Timetable of Little Bee							
Day	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**	**Sunday**
Place	Library	Hospital	School	Garden	Post Office	Football Court	Garden
	OR	**AND**	**AND**	**OR**	**AND**	**OR**	**AND**
Place	School	Supermarket	Football Court	Supermarket	Library	Police Station	Bakery

Figure 5. A card to facilitate the learning of coding education

As reflected in the learning package, the participants demonstrated their ability to design learning activities based on what they learned in the workshops. The design of the learning activities that required children to actively manipulate the Bee-Bot for learning aligned with the learning theory of active engagement suggested by Dewey (1938) and Piaget (1947/1960). The activities focused on the learning of logical concepts and the development of a sequential program. The design included a clear learning and teaching process. The participants made adjustments to the size of the mat from 4x4 (Figure 3) to 3x3 (Figure 4). They opined that a mat of smaller size could reduce the difficulties of composing instructions and that was more suitable to the cognitive development of children.

Conclusion

Coding skill has been emphasized in recent years in the STEM initiative. Preliminary results in this study suggests that pre-service early childhood teachers were able to master fundamental coding skill and they were able to design coding education activities for children after receiving some training. However, since a small number of pre-service teachers participated in this study, another study in a larger scale may be required. Nevertheless, this study highlighted the need to consider appropriate learning contents for kids and an urgent need to provide teacher training on coding education.

Moreover, the researcher suggested that the learning of logical concepts and a sequential program are appropriate for young children. This study also contributed to the field by providing an elaboration of an approach to implement teacher training in an early childhood education programme for future reference.

References

Au, T. K., Sidle, A. L., & Rollins, K. B. (1993). Developing an intuitive understanding of conservation and contamination: Invisible particles as a plausible mechanism. *Developmental Psychology, 29*(2), 286-299.

Awad, N., & Barak, M. (2016). Learning about STEM in the ICT environment – The case of the sound, wave and communication systems course. *Proceedings of the 11th Chais Conference for the Study of Innovation and Learning Technologies: Learning in the Technological Era* (pp. 1E-7E). Raanana, Israel: The Open University of Israel.

Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. *Learning & Leading with Technology, 38*(6), 20-23.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48-54.

Berk, L. E., & Meyers, A. B. (2016). *Infants and children: Prenatal through middle childhood.* London, UK: Pearson.

Bers, M. U., Seddighin, S., & Sullivan, A. (2013). Ready for robotics: Bringing together the T and E of STEM in early childhood teacher education. *Journal of Technology and Teacher Education, 21*(3), 355-377.

Curriculum Development Council (2015). *Promotion of STEM Education.* Hong Kong: HKSAR.

Deitel, P., & Deitel, H. (2015). *Java how to program (late objects).* London, United Kingdom: Pearson.

Department for Education (2014). *National curriculum.* Retrieved from https://www.gov.uk/government/collections/national-curriculum

Department of Education. (2017). *Science, technology, engineering and math: Education for global leadership.* U.S. Department of Education. Retrieved from https://www.ed.gov/stem

Dewey, J. (1938). *Experience and education.* New York, NY: MacMillan.

Ebersbach, A., Glaser, M., & Heigl, R. (2006). *Wiki: Web collaboration.* Berlin, Germany: Springer.

Eck, J., Hirschmugl-Gaisch, S., Hofmann, A., Kandlhofer, M., Rubenzer, S., & Steinbauer, G. (2013). Innovative concepts in educational robotics: Robotics projects for kindergartens in Austria. *Proceedings of the Austrian Robotics Workshop '13* (pp. 19-24). Vienna, Austria: University of Applied Sciences Technikum Wien.

Fernandez, M., Wegerif, R., Mercer, N., & Drummond, S. R. (2001). Re-conceptualizing "scaffolding" and the Zone of Proximal Development in the context of symmetrical collaborative learning. *Journal of Classroom Interaction, 36*(2), 40-54.

Fraenkel, J. R., Wallen, N. E., & Hyun, H. H. (2012). *How to design and evaluate research in education.* New York, NY: McGrawHill.

Gelman, R. (1972). Logical capacity of very young children: Number invariance rules. *Child Development, 43*(1), 75-90.
Gomes, A., & Mendes, A. J. (2007). Learning to program - Difficulties and solutions. *Proceedings of International Conference on Engineering Education - ICEE 2007.* Coimbra, Portugal.
HKSAR. (2017). *2017 Policy address*. Retrieved from http://www.policyaddress.gov.hk/2017/eng/
Hsu, Y.-C., Rice, K., & Dawley, L. (2012). Empowering educators with Google's Android App Inventor: An online workshop in mobile app design. *British Journal of Educational Technology, 43*(1), E1-E5.
Janka, P. (2008). Using a programmable toy at preschool age: Why and how? *Workshop proceedings of International Conference on Simulation, Modeling and Programming for Autonomous Robots*, (pp. 112-121). Venice, Italy.
Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., . . . Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads, 2*(1), 32-37.
Manches, A., & Plowman, L. (2017). Computing education in children's early years: A call for debate. *British Journal of Educational Technology, 48*(1), 91-201.
Ministry of Education. (2017). *Applied learning.* Retrieved from Ministry of Education, Singapore: https://www.moe.gov.sg/education/secondary/applied-learning
Ng, W. S. (2016). Enhancing the quality of educational website design through assessment for learning strategies. In G. Sreedhar (Ed.), *Design solutions for improving website quality and effectiveness* (pp. 24-51). Hershey, PA: IGI Global.
Oldknow, A. (2009). ICT bringing mathematics to life and life to mathematics. *Electronic Journal of Mathematics & Technology, 3*(2), 137.
Osman, K. (2014). Editorial. Science technology, engineering and mathematics (STEM) education in Malaysia. *Eurasia Journal of Mathematics, Science & Technology Education, 10*(3), 153-154.
Park, I., Kim, D., Oh, J., Jang, Y., & Lim, K. (2015). Learning effects of pedagogical robots with programming in elementary school environments in Korea. *Indian Journal of Science and Technology, 8*(26),1-5.
Piaget, J. (1960). *The psychology of intelligence* (M. Piercy & D. E. Berlyne, Trans.). Totowa, NJ: *Littlefield*, *Adams & Co*. (Originally published 1947)
Piaget, J. (1971). *Biology and knowledge.* Chicago, IL: University of Chicago Press.
Punch, K. F. (2009). *Introduction to research methods in education.* London, United Kingdom: SAGE Publications Ltd.
Rao, C. N. (2014). Ethics in relation to science and technology education and development. In M. J. Frazer & A. Kornhauser (Eds.), *Ethics and social responsibility in science education* (pp. 1-12). Oxford, United Kingdom: Pergamon Press.
Roberts, W. E., & Matzen, N. (2013). STEM and ICT instructional worlds: The 3D experience, the impact on today's students. *Journal of Education and Learning, 7*(1), 57-62.

Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172.
Rosen, A. B., & Rozin, P. (1993). Now you see it, now you don't: The preschool child's conception of invisible particles in the context of dissolving. *Developmental Psychology, 29*(2), 300-311.
Smith, M. (2016). Computer science for all (Web log comment). Retrieved from https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all
State University of New York. (1997). *Definitions of information literacy.* Retrieved from http://web.plattsburgh.edu/library/instruction/informationliteracydefinition.php
Sullivan, A., & Bers, M. U. (2016). Robotics in the early childhood classroom: Learning outcomes from an 8-week robotics curriculum in pre-kindergarten through second grade. *International Journal of Technology and Design Education, 26*(1), 3-20.
Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes.* Cambridge, MA: Harvard University Press.
Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Acknowledgement

This project was supported by The Education University of Hong Kong.

Author Details

Wing Shui Ng
ngws@eduhk.hk