

Computational thinking – Essential and pervasive toolset

Erni Marlina Saari^{1*}, Gail Hopkins²

¹Faculty of Art, Computing and Creative Industry,

Sultan Idris Education University, Tanjung Malim, Perak, Malaysia.

²School of Computer Science, University of Nottingham, United Kingdom.

*Corresponding author: marlina@fskik.upsi.edu.my

Received: 21 January 2020; **Accepted:** 31 March 2020; **Published:** 02 April 2020

To cite this article (APA): Saari, E. M., & Hopkins, G. (2020). Computational thinking – Essential and pervasive toolset. *Asian Journal of Assessment in Teaching and Learning*, 10(1), 23-31. <https://doi.org/10.37134/ajatel.vol10.1.3.2020>

To link to this article: <https://doi.org/10.37134/ajatel.vol10.1.3.2020>

Abstract

Education 21st century is all about enfolding digital technology. The theme “Higher Education 4.0: Knowledge, Industry and Humanity”, mandated from Malaysia’s former Higher Education Minister Datuk Idris Jusoh. The minister identified that universities have to be trained to adapt and change the curriculum so that graduates are capable to fill in jobs which are yet to arise. This Fourth Industrial Revolution 4.0 as part of the call to revamp the Malaysia higher education system. There are nine Malaysia future-proof skills that has been listed under Ministry of Higher Education module on Framing Malaysian Higher Education 4.0 – future-proof talents. There are creativity and innovation; Holistic Entrepreneurial and Balanced; Resilience; leadership; Compassion and mindfulness; values and ethics; flexibility and adaptability; critical thinking and problem solving and finally communication and language proficiency. The above mentioned points are future-proof skills sets for Malaysian graduates. There are three additional future-proof attributes which are lifelong learners, multiple intelligence and competencies and computational thinking. This qualitative study explored the significant area in the recent digital technology and development hence, it will be one of the crucial knowledge that should be acquired by everyone and anyone not only in Malaysia but in the whole wide world. Technology is moving rapidly and educators have to keep up with this fast pace. This issue attentively allied with the terminology pioneered by Jeanette Wing that is called computational thinking (CT). The interviewed is carried out among eleven Malaysian pre-service teachers and existing teachers under PGCE (Post Graduate Certificate in Education) program in the United Kingdom, to study on their attitudes towards the idea of learning programming. Through the interviews, the researcher was able to record and interpret trainee teachers’ perceptions of the motivation to learn programming. The data were then analysed and categorised before the final codes were determined to get the final output using thematic code analysis technique. The findings show that learning programming is possible for those who had no computing background thus answered one of the aim of this study that CT skills could be adopted in any fields and more creators or designers will be established compared to the passive users. There was a rational consideration of the value of learning programming to their professional skills and the studies have asked whether understanding such will raise the participants’ engagement with learning a programming language and thus assist them to acquire CT skill.

Keywords: Computational Thinking, Education, Technology, Programming, Teachers

INTRODUCTION

The government’s target for future generation is for them to be technology savvy. Even though the target are not going to become software engineers/ programmers/system developer, but everyone use computers every day and need to appreciate how to converse with them to most effectively harness their computing power (Shute, Sun, & Asbell-Clarke, 2017). Successful communication along these lines is called computational thinking (CT). A main dilemma in promoting the teaching of programming and computational thinking (CT) to primary or secondary school teachers in Malaysia is that most of such



teachers have no theory of how to program and obviously are not motivated to learn programming. CT vision inspired by Wing (2006), is a fundamental skill used by everyone by the middle of the 21st century (such like reading, writing, and arithmetic). As Yadav (2011) stated in his article that CT involves designing systems and understanding human behaviour by drawing on the concepts fundamental to computer science (Wing, 2006). CT had been taught extensively to most of the development countries like United State of America, United Kingdom and Finland. For instance, the purpose of CT is to help breaking down complex problem into smaller problems (Brennan & Resnick, 2012; Wing, 2008; Yadav, Hong, & Stephenson, 2016b). CT is currently also part of the vision in Malaysian Education which consents for the development to be in the same level as other developing counties as mentioned above (Ministry of Higher Education, 2018). It has been integrated in the school curriculum for primary and secondary school. If CT will be used everywhere, then it will touch everyone directly or indirectly. If computational thinking is added to the series of thinking abilities, then how and when should people learn this kind of thinking and how and when should we teach it? (Wing, 2008). This integration is mentioned in the Malaysia Education Blueprint of Pelan Pembangunan Pendidikan 2013-2020. Programming is important in developing the interest of students towards Science, Technology, Engineering and Mathematics (STEM) concept (Ministry of Education (MOE), 2013; 2015). It stimulates students to build, develop and create something based on their creativity in entertaining and exciting ways. Programming also helps the students in learning other subjects as the students can visualize the problem (Kalganova, 2001) in addition and multimedia are designed in assisting the learning process (Novianti & Suparman, 2019) thus enhance their problem solving skills. The main objectives of this study were for participants to prepare a fundamental lesson task for their learners using Flash Action Script and to look at their motivation.

LITERATURE REVIEW



Programming Is Hard



Programming course is seen as a nightmare that has arisen, and the student is strong to see how they can avoid programming in the future (Jenkins, 2001). Programming is a very useful skill and can be a rewarding career (Huang, Ho, & Chung, 2008). Programming refers to an action which uses programming language to teach the computer to complete something and understand the command provided (Ridzuan & Jawawi, 2016). Programming subject was introduced to primary school students at the age of 5 years old while in Malaysia, students were introduced to programming at the age of 16 as an elective subject. It would be a hard burden for novice programmers who are in the initial stage of programming learning. There's an argument that using ad hoc programming languages decreases student motivation, since the programming skills developed cannot be used elsewhere. Perhaps the answer is "It's the programming environment—it needs to support learning the skills of expert programmers." (Guzdial, 2003). Resnick expressed in the interviews about computing being a medium for self-expression: "programming, like writing, is a means of expression and an entry point for developing new ways of thinking" (Bocconi et al., 2016).

Motivation to Learn Programming?

Numerous learning environment has been developed and introduced in motivating people from different age level to learn programming. Different learning approach should be directed based on the age level competency of the students may differ. There is a range of tools offered to teach programming, many of them are either designed for children or require some basic knowledge upon which to build. The trend of using CT in research in all fields is already happening including graduate students (Wing, 2008).

CT and related concepts (for instance coding, programming, algorithmic thinking) have been supported by educational stakeholders as skills that are as fundamental for all as numeracy and literacy. For adults who are new to the domain, teaching programming can be challenging due to having to build CT and programming skills with a teaching method or tool which is age appropriate, and due to the fact





that often there is a fear of computing and programming seen in these learners (Radošević, Orehovački, & Lovrenčić, 2009; Seiter & Foreman, 2013; Tondeur et al., 2012). Teaching a practical discipline like programming requires the teacher to persuade students to agree to practice by engaging in writing programs (Jenkins, 2001a). It has been acknowledged that, in particular, it is not easy to motivate non-technical students to learn programming because programming is not a primary interest of theirs (Takemura, 2007). Boulton (2008) stated that “there needs to be an awareness that younger students of full-time compulsory school age do need training in using e-learning materials and developing independent learning skills”. Such training, however, needs to motivate learners and for adults many tools designed to teach programming may not motivate in the same way that they do for children (E.M. Saari, P. Blanchfield, & G. Hopkins, 2016).

Educators Become Creative Designer Instead of Passive Users

Realizing the significance of introductory programming at school as early as primary school students, a study was conducted among pre-service teacher to identify the suitable programming teaching tools. Various available programming tools were introduced in order to teach programming to the students such as Scratch and Storytelling Alice. These tools provide great help in learning programming by showing visible output to the students.

Scratch

Scratch (Figure 1) is aimed specifically at young learners and provides a valuable resource in schools (Brennan & Resnick, 2013; Resnick et al., 2009). Its effectiveness like Storytelling Alice, however, depends on the learner and their learning style. If a learner cannot see the value in what they are learning then motivating them to use the tool can become a problem (Chang et al., 2012; Lee and Lee, 2014). In brief, Scratch can be easily used by novice programmers and lets users create interactive, media-rich projects. In the website of the software, users can upload projects media and scripts, animated stories, games, online news shows, book reports, greeting, cards, music videos, science projects, tutorials, simulations and sensor-driven art and music projects (Vaca-Cardenas et al., 2015). However, teachers on the whole are less motivated by this and indeed the majority, like those in the experiments we have undertaken, find little motivation to learn to program from these initiatives (E.M. Saari, P. Blanchfield, & G. Hopkins, 2015; E.M. Saari, et al., 2016). Figure 2 shows the interface screen of a Blockly. Blockly in a browser allows web pages to include a visual code editor for any of Blockly's five supported programming languages, or your own. In Blockly Games, users can solve a maze using Blockly's editor on the right. Google and the MIT Media Lab's Scratch Team are grouped on the expansion of a new generation of graphical programming blocks, called Scratch Blocks. Scratch Blocks builds on Google's Blockly technology and the Scratch team's expertise in designing creative interfaces for young learners.



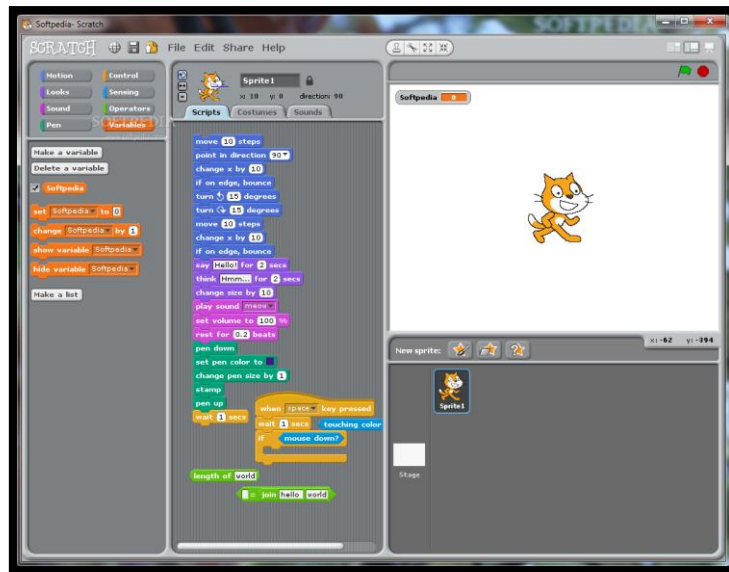


Figure 1. Scratch Interface

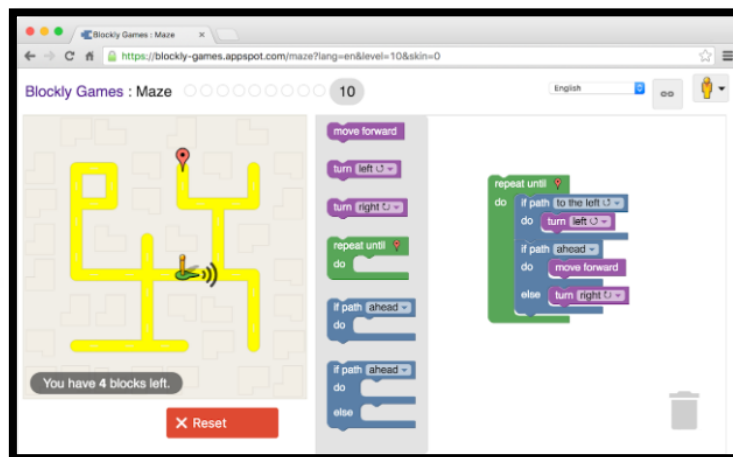


Figure 2. Blocky Interface

Storytelling Alice

Storytelling Alice (SA) is aimed at teaching fundamental programming skills such as variables, looping and conditionals. It was originally designed to engage middle school girls aged 12 years and above and develop their digital fluency and CT skills (Daily, Leonard and Jörg, 2014). Learners construct programs by dragging and dropping code elements, thus removing the possibilities of making syntax errors, whilst the animations allow them to see semantic errors as they occur. Researchers also use Alice as a tool to support the development of algorithmic thinking, problem solving, and event processing (Daily, Leonard, & Jörg, 2014; Utting, Cooper, & Kölling, 2010).



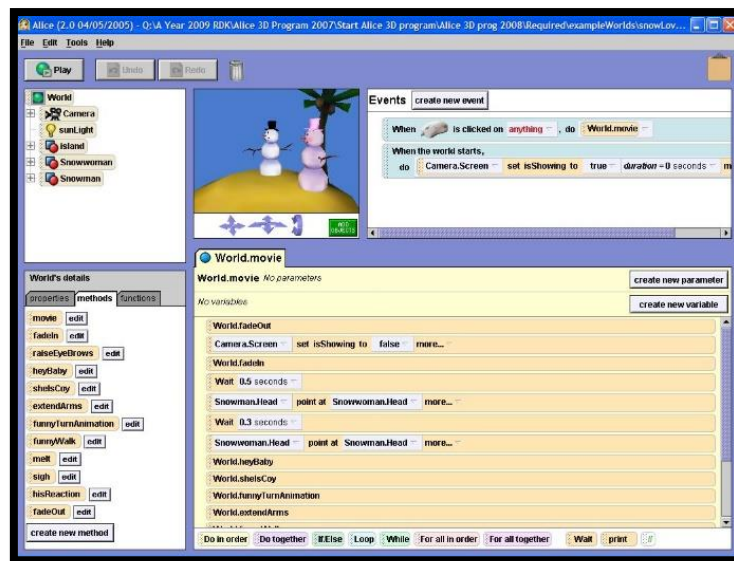


Figure 3. Storytelling Alice Interface

In Malaysia, teaching CT is still a relatively new development, formally introduced at the time of writing, in 2017, despite the fact that learning CT through robotics has been practiced at selected schools prior to this. These selected schools (Sekolah Bestari) have been using robotics to teach students since 2003. However, cost has played a large part in the success of this and has meant that implementing more widespread teaching of CT has been impaired. Teachers or schools that have adequate facilities (e.g. computer labs) are normally in urban areas or have been selected as high-performing schools and have been given the provisions to teach CT using the latest technology (Abas, 2016; Azlee, 2016).

METHODOLOGY

In order to investigate the difficulties observed in the Malaysian trainee teachers and the role of the proposed tool in teaching CT, a group of Malaysian test subject was recruited for this qualitative study. They are original eleven Malaysian trainee classes studying on a PGCE course in the UK. They had no background experiences in the use of software development or computing background. The study was subject to the ethics approval process. All participants were informed of the nature and purpose of the study before partaking and were aware of the recording methods that were to be used. Prior to using the pilot tool each subject was interviewed initially about their attitude to using the Flash environment and other aspects of using technology, particularly the learning of programming tools including the use of Scratch. The lessons were presented one-on-one but otherwise in a way similar to the original group (lecture, PowerPoint presentation and subsequent practical exercise). Each subject's comments and progress were recorded during their performance. The data captured (using Camtasia Studio software) included length of time to complete actions, evidence of enthusiasm for the task and number and types of errors made. After the lesson the subjects were once again interviewed about their attitudes towards use of Flash tools. The recordings were transcribed for analysis. A set of thematic codes was developed to use in determining the level of motivation and engagement with the activity.

FINDINGS

Eight pre-service teachers and three in-service teachers took part in the experiment. The expected output from the completed task was also shown to the participant before they began the task (see Figure 1). The purpose of this was to allow the participant to have a clear understanding of what to do and to enhance their motivation to do so (Lin, 2012; Tsukamoto, 2012). The expected output for this study: produce a simple scene for the artefact; add controls and add actions; introduce more complex actions.



The purpose of such steps was to allow the extent of the capability of action scripting to be explored, but also to allow a number of CT skills to be explored (Dorling M., 2014):

- The task must be broken down into a suitable set of sub tasks (Decomposition).
- Action generalization can be seen as activities have many similarities (Generalisation).
- Algorithms are seen being developed in the Action Script codes (Algorithmic Thinking).
- These can then be extended as the complexity of the task increases (Abstraction).
- Debugging was discussed by asking respondents to solve the problem thus errors were detected and solved (Analysis).

Flash Action Scripting is considered to be a language that is really beneficial for beginners (Crawford and Boese, 2006; Sands, Moukhine and Blank, 2008; Fitriza, 2012). One of the best ways to learn how to write Action Scripts is to read the code of an existing Flash project and then reproduce and perhaps alter this code in one's own project (Sorapure, 2006), thus learning by doing. Through the process of learning how to write programs it was envisaged that CT skills could be detected. Use of animation and visuals can greatly aid the teaching of new concepts in programming (Routledge, 2007) and Flash additionally provides a frame-based structure which makes it easier for inexperienced programmers (E.M. Saari, Blanchfield, & Hopkins, 2016) to create ergonomic content (Reimers and Stewart, 2007).

Table 1. Positive and negative codes from participants before the study.

Positive Quotes	Code
"I think this software really helps me in understanding the concept now..." (ET1)	Confident, Affordance
"...by learning this software we can produce a good teaching module for our students..." (TT5)	Motivation, Affordance
"I want to learn this programming more and more so that I will come out with my own teaching modules soon!" (TT1)	Motivation, Affordance, Interest
"I knew that Flash can do interesting stuff..." (TT6)	Interest
"I am looking forward to learn and try something new today..." (TT1)	Motivation, Enjoyment
"I am ready to learn new lesson today!" (TT8)	Motivation, Engagement
Negative Quotes	Code
"I am not that confident because I am sure it will be very tough..." (ET3)	Fear, Difficult
"I'm sure it will be harder and so sorry if I can't finish the tasks for today..." (ET2)	Difficult, Frustration
"Are you sure today's lesson is not difficult for me?" (TT5)	Fear, Difficult
"More complex programming? I am not sure whether I can do it" (TT8)	Fear, Lack of Motivation

Table 2. Positive and negative codes from participants after they had completed their tasks

Positive Quotes	Code
"I feel more confident now..." (T1)	Confidence
"Great lesson learnt today!" (T2)	Enjoyment, Motivation
"I really enjoy the lesson, even though it is quite complicated and hard but..." (TT8)	Enjoyment, Engagement, Motivation
"I can say I am in love with Flash Action Script!" (TT2)	Motivation, Enjoyment
"I can now be a developer at my school..." (TT3)	Affordance, Motivation
"I am speechless because I wasn't thinking I can program at last..." (TT2)	Motivation, Affordance
"...exciting software..." (TT6)	Interest, Engagement, Enjoyment



Negative Quotes	Code
"Sometimes the scripts still confuse me..." (T2)	Confusion, Frustration
"I have mixed up with some scripts..." (TT7)	Frustration, Difficult, Confusion

Even though intrinsic motivation of a student has been recognized as a critical success factor, sensible design of instruction can help boost motivation amongst students (Balan & Kalavally, 2012; E.M. Saari et al., 2016; E.M. Saari et al., 2015).

DISCUSSION

Lin (2012) focuses that educators can become creative courseware designers and certain environments have been used by educators to work on this. Technology in education can provide different teaching methods. However, if in-service teachers and pre-service teachers appreciate taking these approaches and relish using technology in support of this then teaching will be made easier. It has been established that to use technology in teaching ideally requires pre-service teachers to not only learn how to use the technology but to enjoy doing so (Boulton & Hramiak, 2013) and, in turn, motivating pre-service teachers, and particularly those from a non-technical domain is crucial. The other challenge is convincing students that they are skilful of programming to solve problems (Martins et al. 2010). Hence, the Malaysian Ministry of Education aims to cultivate the standard and quality of education and technology among educators in Malaysia for the Education 4.0 would be succeed. If a programming environment motivates users then they are more likely to keep working (Wang, Wang and Liu, 2014) but only if they perceive that ultimately they have the capability to overcome the task they have been given.

Correct use of CT involves a programmer breaking down a problem into smaller problems (decomposition) and understanding how to solve those problems algorithmically (algorithmic thinking). Programmers also need to be able to abstract from one problem to another; in other words to understand how one problem might stem from another; also programmers should know how to debug a problem and solve it (analysis) and to generalize so that they can apply a previous solution correctly to another problem (E.M. Saari et al., 2016).

In this paper it became noticeable that CT was not being used in many of the subjects. One set of behaviours that was frequently observed in the lessons was that of copying and pasting code rather than understanding the given algorithm and adapting it to the new problem. This clearly showed that wrong – non-Computational Thinking had taken place.

The opportunity to see algorithms developed was clearly useful in stimulating thinking amongst participants although one of the in-service teachers did drop out of the process. At this point the basic needs for writing the self-help tool had been identified and the next stage was to develop and test this tool. The development and testing of this are the subject of the research to be taken.

CONCLUSION

The Flash Action Script software engaged participants and levels of motivation to try and program seemed improved after the participants had used it for a while. Based on the lessons and the responses of participants - Flash Action Scripting could enable those without a computing background to develop CT skills. Identification of where participants were going wrong and using incorrect Computational Thinking was possible and this was most apparent by their copying and pasting behavior showing that they could not think algorithmically or generalize to other problems. This research is aimed at adapting Computational Thinking (CT) skills and not just learning how to create content by integrating multimedia elements. Instead the aim of creating that content is to connect an individual's intrinsic motivation to be able to develop their own learning material, which they themselves can benefit from using in their lessons. Students could apply computational concepts learned from Scratch to specific

text-based programming languages, which is quite promising. In any case, the long-term retention of CT skills, along with the application of CT skills to other contexts and domains, is under-researcher. Additionally the resulting development of CT skills and ability to program can also be of benefit to their teaching profession, and this intrinsic motivation was perceptible in several of the participants.

REFERENCES

- Abas, A. (2016). Computational thinking skills to be introduced in school curriculum next year. *News Straits Times*, p. 164732.
- Azlee, B. Y. A. (2016). Lack of info on coding in schools prompts concern among educators.
- Balan, P., & Kalavally, V. (2012). Enhancing Student Motivation in Process Control via Interactive Learning Tools. ... *for Education (T4E), 2012 IEEE Fourth ...*, 176–179. <https://doi.org/10.1109/T4E.2012.38>
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., Engelhardt, K., Kampylis, P., & Punie, Y. (2016). *Developing Computational Thinking in Compulsory Education. Implications for policy and Practice. Proceedings EdMedia 2016*. <https://doi.org/10.2791/792158>
- Boulton, H., & Hramiak, A. (2013). Cascading the use of Web 2.0 technology in secondary schools in the United Kingdom: identifying the barriers beyond pre-service training. *Technology, Pedagogy and Education*, 23(2), 151–165. <https://doi.org/10.1080/1475939X.2013.802994>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. ... *of the 2012 Annual Meeting of the ...*. Retrieved from <http://pdf.thepdfportal.com/PDFFiles/26416.pdf>
- Brennan, K., & Resnick, M. (2013). Stories from the scratch community: connecting with ideas, interests, and people. In *Proceeding of the 44th ACM technical symposium on Computer science education - SIGCSE '13* (p. 463). <https://doi.org/10.1145/2445196.2445336>
- Daily, S., Leonard, A., & Jörg, S. (2014). Dancing alice: exploring embodied pedagogical strategies for learning computational thinking. *Proceedings of the 45th ...*, 1–6. Retrieved from <http://dl.acm.org/citation.cfm?id=2538917>
- Guzdial, M. (2003). Programming Environments for Novices Specializing Environments for Novices Logo and its Descendants : The Goal of Computational Literacy, 1–24.
- Huang, C., Ho, P.-C., & Chung, S.-M. (2008). Computer Game Programming Course for Art Design Students by Using Flash Software. *2008 International Conference on Cyberworlds*, 710–713. <https://doi.org/10.1109/CW.2008.120>
- Jenkins, T. (2001). TEACHING PROGRAMMING – A JOURNEY FROM TEACHER TO MOTIVATOR. *Annual LTSN-ICS Conference in London*.
- Kalaganova, T. (2001). Improving students motivation by means of multimedia. Retrieved from <http://dspace.brunel.ac.uk/handle/2438/2520>
- Novianti, N., & Suparman, S. (2019). Educational game design to improve reasoning skills. *Asian Journal of Assessment in Teaching and Learning*, 9(2), 1–8.
- Radošević, D., Orehovački, T., & Lovrenčić, A. (2009). New approaches and tools in teaching programming. *Proceedings of 20th Central ...*, 49–57. Retrieved from http://bib.irb.hr/datoteka/427643.CECHS2009_Radosevic_Orehovacki_Lovrencic.pdf
- Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., ... Silver, J. (2009). Scratch. *Communications of the ACM*. <https://doi.org/10.1145/1592761.1592779>
- Ridzuan, F., & Jawawi, D. N. A. (2016). Evaluation of Computer Programming Teaching Tools for Secondary Schools Students. *International Journal of Software Engineering and Technology*, 03(1), 1–9.
- Saari, E.M., Blanchfield, P., & Hopkins, G. (2016). *Computational thinking: A tool to motivate understanding in elementary school teachers. Communications in Computer and Information Science* (Vol. 583). https://doi.org/10.1007/978-3-319-29585-5_20
- Saari, Erni Marlina, Blanchfield, P., & Hopkins, G. (2015). Learning computational thinking through the use of flash action scripts: Preparing Trainee Elementary School Teachers for Teaching Computer Programming. *CSEDU 2015 - 7th International Conference on Computer Supported Education, Proceedings*, 2, 75–84. <https://doi.org/10.5220/0005442600750084>
- Saari, Erni Marlina, Blanchfield, P., & Hopkins, G. (2016). *Computer Supported Education*. (S. Zvacek, M. T. Restivo, J. Uhomobhi, & M. Helfert, Eds.), *Computer Supported Education* (Vol. 583). Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-29585-5>
- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research - ICER '13*, 59. <https://doi.org/10.1145/2493394.2493403>

- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking ARTICLE INFO. *Educational Research Review*. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Takemura, Y. (2007). Analyzing the Motivation of the Students in the Art Faculty for Learning Programming Department of Social Welfare and Psychology, (Icalt), 0–2.
- Tondeur, J., van Braak, J., Sang, G., Voogt, J., Fisser, P., & Ottenbreit-Leftwich, A. (2012). Preparing pre-service teachers to integrate technology in education: A synthesis of qualitative evidence. *Computers & Education*, 59(1), 134–144. <https://doi.org/10.1016/j.compedu.2011.10.009>
- Utting, I., Cooper, S., & Kölling, M. (2010). Alice, greenfoot, and scratch--a discussion. *ACM Transactions on ...*, 10(4), 1–11. <https://doi.org/10.1145/1868358.1868364>
- Vaca-Cardenas, Azucena, L., Bertacchini, F., Tavernise, A., Gabriele, L., Valenti, A., ... Bilotta, E. (2015). Coding with Scratch: The design of an educational setting for Elementary pre-service teachers. *Proceedings of 2015 International Conference on Interactive Collaborative Learning, ICL 2015*, (September), 1171–1177. <https://doi.org/10.1109/ICL.2015.7318200>
- Wing, J. M. (2006, March 1). Computational thinking. <https://doi.org/10.1145/1118178.1118215>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>
- Yadav, A., Hong, H., & Stephenson, C. (2016a). Computational Thinking for All : Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*, 10–13. <https://doi.org/10.1007/s11528-016-0087-7>
- Yadav, A., Hong, H., & Stephenson, C. (2016b). Computational Thinking for All: Pedagogical Approaches to Embedding 21st Century Problem Solving in K-12 Classrooms. *TechTrends*, 60(6), 565–568. <https://doi.org/10.1007/s11528-016-0087-7>