FACULTY OF SCIENCE

School of Computer Science

TOOLBOX FOR ADOPTING COMPUTATIONAL THINKING THROUGH LEARNING FLASH

ERNI MARLINA SAARI

Thesis submitted to the University of Nottingham UK

For the degree of Doctor of Philosophy

November 2018

Abstract

1. The need for teachers of Elementary School children to learn to program or rather to understand the Computational Thinking behind programming has been accelerated in many countries by the mandated teaching of programming in the Elementary School context. Many steps have been taken in order to create awareness of this issue, such as the Computing At Schools initiative (CAS) which is established in the UK. CAS aims to support teaching in computing and connected fields in UK schools. Moreover, in the USA the Computer Science Teachers Association (CSTA) was established to meet the purpose of informing and advising about the current development of computational thinking and to investigate and disseminate teaching and learning resources related to computational thinking. In Singapore research has been conducted by the government agency Infocomm Development Authority of Singapore (IDA) whereby the major goal is to meet the needs in the ICT sector and ultimately to focus and inspire learners about programming.

The research for this thesis involves the development of a training scheme for pre-service teachers that will introduce them to computational thinking through the use of the Flash Action Script Development environment. Flash Action Scripts – amongst several other tools – are used as a tool for creating interactive content and because Flash is one of the premiere tools used to create content for the internet; a tool programmed with Flash looks practically the same in every browser and on every operating system. Flash Action scripts use traditional coding skills but permit the user to see how each piece of code affects the running or execution of the program, allowing the user to have an instant visual understanding of what the code is doing. It is also widely available within university campuses.

A major problem in promoting the teaching of programming and computational thinking to Elementary School teachers is that the majority of such teachers have no concept of how to program and naturally are not motivated to learn programming. Experienced teachers involved in the current study felt that programming was too complicated and thus it was hard to gain fluency in programming. Student teachers who had no previous

experience in programming were, however, easier to get engaged in learning programming principles. Eighty percent of this group found Action Scripting a useful tool to understand basic programming and scripting. The need to teach programming will motivate most but to learn through a tool that can be seen to have intrinsic value in their role as teachers has a greater potential of success. This thesis defines the design and implementation of a tool to use the learning of Flash Action Scripting as a motivational mechanism for pre-service teachers. The intrinsic value to them is intended to be utilisation of the learned Action Scripting skills to produce their own teaching material. Initial results indicate an enhanced engagement and motivation to learn to program and improved confidence in doing so. As projected the pre-service teachers had a more positive attitude towards the potential of the learning tool but both they and the in-service teachers had improved attitudes and enthusiasm after the experiment. The results show that both pre-service and in-service teachers can be trained to be designers and producers of digital courseware in the previous absence of computational thinking skills and definitely they can acquire skills in computer programming such as Flash Action Scripts.

Acknowledgements

In the name of Allah, the Most Gracious and the Most Merciful

Alhamdulillah, an abundance of praise is due to The Rabb, The Merciful for His nurture, excessive mercy and blessing in this thesis completion.

In the completion of this thesis, I owed many people in different ways. First and foremost I would like to render my gratitude to the management of the Sultan Idris Education University (UPSI) for awarding me the scholarship and study leave and The Ministry of Higher Education Malaysia for the full sponsorship. Throughout my studies here, I am truly indebted and thankful for all the support that I have received. In general like to thank the following people:

Special thanks to both of my supervisors, Associate Professor Dr. Peter Blanchfield and Associate Professor Dr. Gail Hopkins, for guiding me and relentlessly advising me without fail, assisting and pointing me in the right direction. I have learned so much from both of them and I will always cherish of all the lessons, meetings and advice given.

My appreciation also goes to all my friends for helping me and make my PhD life enjoyable.

To my ever loving family back home, thank you for the prayers for my success in this task entrusted upon me. My beloved mother, Zaiton and all my siblings, I hope I have made you proud and essentially thank you for all the support. And I know, I would have made bapa proud too, Al-Fatihah.

Words and numbers could not represent the amount I owed to my own family, my children Nurdayana Batrisya, Danish Irfan, Danish Firhan and Nurdamia Aleesya. Finally, if anyone deserves a multitude of recognition and gratitude, it will be my dearest husband, Mohamad Zamri Abdul Aziz who has not only been the heart to the family, but also my inspiration. Thank you for the love, the humor and most importantly the constant assurance that 'I can do this' for without your love and sacrifices, PhD is only a dream afar. I dedicate this thesis to you.

Declaration of Authorship

I, Erni Marlina Saari, declare that the thesis entitled "Toolbox for Adopting Computational Thinking through Learning Flash Action Scripts" and work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that this work was done wholly or mainly while in candidature for a research degree at this University;

Where I have consulted the published work of others, this is always clearly attributed;

Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;

I have acknowledged all main sources of help;

Where the thesis is based on work done by myself, I have made clear exactly what was done by others and what I have contributed myself;

Part of this work have been published as:

Saari, E.M., Blanchfield, P. and Hopkins, G., 2015. Learning Computational Thinking through the Use of Flash Action Scripts: Preparing Trainee Elementary School Teachers for Teaching Computer Programming *In: CSEDU 2015, The 7th International Conference on Computer Supported Education, Proceedings* 2: 75–84.

Saari, E.M., Blanchfield, P. and Hopkins, G., 2016. Computational Thinking - A Tool to Motivate Understanding in Elementary School Teachers. Pp. 348–364 in *Computer Supported Education*, *Chapter of Book*, edited by Markus Helfert Susan Zvacek, Maria Teresa Restivo, James Uhomoibhi. Switzerland: Springer International Publishing Switzerland.

Signed:  November 2018

Table of Contents

05-4506832 pustaka.upsi.edu.my Perpustakaan Tuanku Bainun Kampus Sultan Abdul Jalil Shah PustakaTBainun ptbupsi

05-4506832 pustaka.upsi.edu.my Perpustakaan Tuanku Bainun Kampus Sultan Abdul Jalil Shah PustakaTBainun ptbupsi

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

The need for more programmers is now serious and software companies and governments around the world have become convinced that a change in the way technology is taught in schools is needed. It is felt vital that children begin to learn to program and more essentially think like programmers from an early age. An important issue arises in how to introduce programming at an early age: how can teachers without a background in computing learn how to teach programming? A common sense answer to this might be that they should first learn to program themselves but how should this be tackled? Importantly, what will motivate them to learn?

The work in this thesis focuses specifically on tackling this problem in a Malaysian context though much of what needs to be done will be relevant in a broader context. Educators the world over are trying to respond to these challenges. The situation in Malaysia needs specific responses relating to the culture of education in that context but many of these problems will also be true elsewhere. Of specific interest is tackling the expectations of continual professional development in Malaysia. At the outset of this research it was expected that there would be much to learn from the experience in other places and particularly in the UK where work was already underway in tackling this.

The term computational thinking (CT) is a way of expressing the sort of thinking needed by programmers. (Please Chapter 2 Section 2 where the definitions of CT are explored in detail.) In introducing this concept to a broad audience of those with an interest in computing Jeannette Wing (2006) tried to classify CT as ways of thinking that went beyond programming specifically and looked at it as a way of approaching problem solving more generally. The idea was that every student should be given the chance to engage with a computational thinking independent of their area of study (J. M. Wing 2006a). CT involves more than learning to program, but many of the concepts in CT are

exercised during programming (Duncan, Bell, and Tanimoto 2014). CT can be exploited in many situations and can lead to better approaches to problem solving throughout life. By Wing's definition CT is human thinking (problem solving by humans rather than just machine thinking) and involves taking an approach to solving problems, designing systems, applying tools, techniques and understanding of human behaviour that draws upon concepts essential to computing (J. M. Wing 2006a, 2008; Grover and Pea 2013). CT does not mean that all students must master the details of a complex programming language but it is essential for students to understand the ideas of programming, to appreciate the difficulty of expressing a solution in a form that can be interpreted by a machine, and to grasp the concepts of debugging (Curzon et al., 2009). As such it creates an important focus on the concept of thinking about problems in a way that leads to solutions that may be implemented in a computing device (or perhaps a teaching tool as will be talked about in this thesis). Making CT manageable for teachers or students may mean it can be possible to teach computing more effectively. Whilst computing education researchers explore how humans come to understand computing, and how to improve that understanding, the research in this thesis studies the idea that learning to think computationally will enhance pre-service teachers' abilities to learn how to teach programming.

The concept of Computational Thinking has thus become a topic of discussion (Dorling M., 2014; Denning, 2009; J. M. Wing, 2006a) and the role that this plays in developing the correct thinking for learners to become programmers is clear (Dorling M. 2014; Csizmadia et al. 2015). It is not expected that teachers will themselves become programmers but they need to acquire the skills to teach programming and this is the case in countries throughout the world.

Malaysia's vision to become a developed nation by the year 2020 (Bernama 2008) has placed science and technology at the top of the list of subjects in which to excel. There is no specific subject on computer programming that has been developed for primary school students in Malaysia. Learning has instead been more focused on use of application software (L. Lezam et al., 2015). Computer programming is to be incorporated into the national curriculum of primary schools in Malaysia from 2017 according to the MDEC (Multimedia Digital Economy Corporation) CEO Datuk Yasmin Mahmood,

indicating the importance that is now being placed on coding skills (Azlee 2016). "Educators should be made aware that teachings of programming are not only confined to acquiring the skill of programming but acquiring this skill could also increase the understanding of more fundamental subjects like language, mathematics and science" (Husain, Kamal, and Ibrahim 2017). This was also announced by Malaysia's Ministry of Education Director-General Khair Mohamad Yusof, who confirmed that Computational Thinking (CT) would additionally form an important part of the curriculum from January 2017 (Azlee 2016; A. Asohan 2016). However, this is not a completely new directive; in 1988 the former Education Minister Anwar Ibrahim announced that computing skills and programming would be part of the curriculum by 1990 (A. Asohan 2016). However, political issues prevented this objective from being realised at the time (A. Asohan 2016).

Whilst there are defined plans to raise the standard and quality of ICT education in Malaysia (Dewitt, Alias, and Siraj 2016; Suliman, Hawari, and Othman 2011; Khambari et al. 2010; A. Asohan 2016) and it is accepted that teaching coding in schools is important (Bernama 2017) awareness of CT is still low when compared with countries such as the USA and the UK. Challenges exist around the fact that learning to programme is not just about having Computer Science as a subject on the curriculum (Grover and Pea 2013; Brown and Kölling 2013; Gretter and Yadav 2016). It is important that CT skills, such as learning how to dissect problems and formulate solutions, form a basis for good programming (Ebrahimi et al., 2013; Kelleher et al., 2007; Sentance & Csizmadia, 2016). Additionally, any programme of education would need to involve not only educating students, but also educating teachers to teach coding skills. Presently, in Malaysia, there has been little in the curriculum which has addressed this and little provision for pre-service and in-service teachers to learn to teach programming. Teachers have recently shown concern regarding the programme to integrate coding into the National School Curriculum stating that they had not been informed of the programme early enough (Azlee 2016). Malaysia's National Union of the Teaching Profession President Hashim Adnan has reinforced this view and emphasised the importance of training the next generation of teachers in a systematic way before the introduction of teaching coding in schools. He stated that without a proper foundation to the programme, it would fail, but that no preparation appeared to be taking place (Azlee 2016). It is apparent that a lot of

work needs to be done if the country is to achieve its goal of a full roll-out of ICT provision in all schools by 2020.

A study that was conducted at one of the top education universities in Malaysia highlighted Malaysian students lack of problem solving skills (Dewitt, Alias, and Siraj 2016). The Malaysian Education Blueprint from preschool to post-secondary, and for higher education has noted that Malaysian students need to develop thinking skills in order to be prepared for their future jobs (Ministry of Education (MOE), 2013; 2015). In January 2017, computational thinking was made part of the reviewed Standard Curriculum for Primary Schools (KSSR) for Year One pupils – making Malaysia the first country in Asean to introduce it into the national syllabus. Deputy Education Minister Datuk Chong Sin Woon added that the programme had been developed with the help of MDEC and that the syllabus was benchmarked against courses offered by Britain's Computing at School as well as the United States' Computer Science Teachers Association (CSTA). "Computational thinking is the ability to dissect problems and formulate solutions by drawing from concepts in computer science." (https://www.digitalnewsasia.com) said MDEC vice-president for talent and digital entrepreneurship Sumitra Nair (http://www.digital-economy/computational-thinking-come-fore-malaysian-schools). Malaysia's education ministry director general Tan Sri Dr Khair Mohamad Yusof said this is not just about having computer science as a subject. Malaysia currently must begin educating students to have computational thinking skills, linking the move to also helping Malaysia achieve its goal of being in the top 30% of the 2025 Programme for International Student Assessment (PISA) assessment. MDEC and the Ministry of Education are strongly concerned that the responsibility to educate, excite and enrich students cannot just start and end in school, which is why the private sector and academia have been tied in through the #mydigitalmaker movement. #Mydigitalmaker is a public-private-academia partnership scheme to move away from being digital users and towards a nation of digital developers. According to MDEC, the private sector and academia will continue to nurture and groom students outside of the classroom through the introduction of Digital Maker Clubs in schools and Digital Maker Hubs in communities. There are six such Maker Hubs in Malaysia at present and this is expected to rise to 14 by the end of 2018.

Students should be inspired in order to help them succeed in any academic assignment. Jenkins and Davy (2002) stated that the goal of a teacher should not only be to transfer knowledge to their learners but also to motivate them to learn and to help them realize the benefits of learning. Learners should be encouraged to explore and apply the concepts to other aspects of learning. However nowadays, it is hard to nurture and maintain students' motivation to study programming merely with the classical programming exercises designed to perform a sequence of calculations.

Over the past 25 years the understanding of the existence of numerous learning styles has brought increasing attention to the idea that students learn in diverse ways and that the purely verbal approach to teaching does not work for every student or even most students (Hawk 2007). The importance of programming skills is becoming more evident in the UK as they have been made core curriculum objectives for school students. In order to teach programming there needs to be a route for teachers to develop an understanding of the underlying concepts. However, it is not necessary to have an understanding of computer programming to design and produce digital material for teaching (Werner et al. 2012). The University Pendidikan Sultan Idris (UPSI) in Malaysia is typical of teaching universities in Malaysia. Personal experience teaching there in early 2011, was the starting point for the current research. It was realized that what a group of students (that did not have any prior computing skill yet took a multimedia programming course) needed was to be provided with the right methods of learning - so they could practice success, which would inspire them to continue studying multimedia programming after leaving the class. In UPSI Adobe Flash Action scripting was being taught to students including those without a computing background. Action scripting allows users to add dynamicity to their multimedia presentations and this has prompted the research for this thesis to investigate whether learning through action scripting can be used to teach underlying programming concepts to teachers while they create their learning material (Lin 2012); in other words, whether it can stimulate the development of CT and therefore an understanding of programming. Some of the components of CT include the following (http://beaver.my): formulating problems in a way that enable the use of computers and other tools to help solve them; logically organizing and analysing data; representing data through abstraction such as models and simulations; automation

of solutions through algorithmic thinking (a series of ordered steps); identifying, analysing and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources and finally, generalizing and transferring this problem solving process to a wide variety of problems. Beaver is suitable for students of a wide range of ages and particularly elementary. Participants do not need prior computer science or programming knowledge. There are 5 categories for students aged 6 to 19 (Year 1 to Form 6). According to Malaysia Deputy Education Minister Datuk Chong Sin Woon, 8 000 teachers in Malaysia underwent training in July 2017 and will be fully equipped to teach computational thinking to students one to two years later. Datuk Chong added that this special training was being conducted with the help of 100 lecturers from teacher training institutes, ministry officers who are British Computer Society-certified master trainers, as well as 90 lecturers from Institut Aminuddin Baki (IAB) (http://www.thestar.com.my/news/nation/2017). Because the term 'computational thinking' is a relatively recent introduction (J. M. Wing 2006a), there is less literature published in Malaysia for instance specifically with this key word. Therefore, the results of the proposed study should help fill the gap between the teaching of programming and computational thinking, specifically in the context of the pre-service and in-service teachers in the Malaysia educational system.

Although programming is a separate skill from CT, the latter draws heavily upon concepts fundamental to computing (Wing, 2008). CT encompasses several skills important for problem solving and can result from studying the nature of computation. It also incorporates creativity, the ability to explain and team work. Effective programming relies on the ability to think computationally, particularly the ability to think logically, algorithmically and recursively (Hu 2011; J. M. Wing 2008; Yinnan and Chaosheng 2012). Specifically, this refers to how a user thinks when producing programs, rather than producing the code itself. Such skills are both important to learn when teaching others to program and also transfer to other problem solving domains in life. For many the whole concept of programming seems foreign and difficult and much work has been done to look at how to get people to learn programming (Crawford and Boese 2006; Martins, Mendes, and Figueiredo 2010). Much that is successful involves people learning through concrete problems where they understand what the result of the process should be. The

internal workings of computers are, however, mysterious to many and so conceiving what is needed to solve a problem through a computer program can be too difficult to imagine and students can be fearful of trying.

In order to motivate students to learn programming many instructors have tried to use games, robots and narrative media hoping that they would impress their students (this is covered in Chapter 2). Jenkins (2001b) highlights the difficulty in obtaining solid conclusions from research on the motivation of students for programming. Students tend to learn in different ways and so wish to use varied teaching resources. There is a variety of tools that have previously been used in teaching programming but these tools generally require some computing background from the students.

The current work has focused on working with teachers who need to learn to program and who desire to learn programming and to develop teaching tools to demonstrate concepts to their students. Electronic artefacts to demonstrate a specific learning area are not always available commercially and those that are available are often expensive and may not be an exact match for the specific needs of the teacher.

The primary aim of this research was therefore to try to establish a way of changing the teachers views on the difficulty and challenges of learning to program for many and a number of research questions were posed, as detailed in the following section.

1.2 RESEARCH QUESTIONS

The discussion so far has set the scene for the main hypothesis of this research. That is in order to meet the desire that children in Malaysia (and elsewhere) learn to program it is first essential that their teachers understand the thinking behind programming and are thus more able to understand how to teach it. At the time this was being conceived the position in Malaysia was particularly difficult. There was little or no incentive for teachers to learn to teach programming. However, the basic belief behind this research was that the need was there and would soon become critical. This belief has been borne out by more recent developments that are making the teaching of programming compulsory at all levels in school education in Malaysia. Experience of teaching trainee teachers had led to the conclusion that many teachers with non-computing backgrounds would be

motivated by learning how to create artifacts using a tool like Flash Scripting. Further this would allow them to have a reason to learn to think computationally for themselves. To investigate the feasibility of teaching programming using a computer-based tool the proposed work needed to be divided into a number of steps. The first step was to investigate more formally the impression given by experience that teachers and trainee teachers were generally not already thinking computationally. Thus the first research question was formulated as:

RQ 1 Are primary school teachers ready to teach programming?

In order to answer each of the questions a number of objectives needed to be attained. In this case:

- To more formally identify attitudes to programming among teachers and trainees in particular.
- To identify the challenges for them in learning to teach programming.
- To determine to what extent they already think computationally and identify ways in which this might be helping or preventing them meeting the challenges of learning to program.

Achieving these objectives would imply a number of things:

First a literature search was needed to look at other research evidence of attitudes of trainee and other teachers to learning to program One aspect of the literature review would be to find ways in which computational thinking were already being measured and thus what would be needed to measure its presence or absence in the teachers. It would also be necessary to survey trainee teachers more formally about their attitudes. This was undertaken at several stages in the research. Previous experience in teaching teachers to use IT tools had suggested that they were generally fearful of learning to program and of using other technology beyond office tools. Thus the second research question was expected to need answering

RQ 2 Can those without a CT background learn to think computationally?

One way in which this would be expected to be promoted would be through motivation. Thus this question implies that many teachers and trainee teachers will be found not to think computationally and probably not be naturally intrinsically motivated to learn. This then implies another objective:

- To understand how teachers might be motivated to learn to think computationally.

Thus part of the literature review would focus on motivation. Previous experience of trainee teachers at UPSI was that learning to use Flash Action scripting was attractive to non-computing students. Thus it was expected that learning this would be a possible motivator to change their attitudes at least. Thus the first intervention was designed to carry on from the initial survey and concentrate on those who had already expressed an interest in taking an elective module in Flash Action Scripting (Flash). The initial survey would then be repeated with the same group of students after exposure to Flash. This would allow a third research question to be posed:

RQ 3 Can we detect where CT is not happening and lead the learner to re-analyse their thinking strategies and consequently learn to think computationally?

Objective:

- To work out how to detect use or lack thereof of the five CT domains: abstraction, algorithmic thinking, decomposition, generalization and analysis.

Should it be possible to teach CT, part of the value in this, and an aim of the research, was to see whether the newly acquired skills could be transferred to computer programming. Thus both in the second and third phases it was the aim to see whether first Flash and subsequently a similar but different software could be programmed effectively by the learners who had been originally unfamiliar with the ideas in programming. Consequently the next research question was:

RQ 4 Can pre-service teachers who have adopted CT demonstrate this by applying it to undertaking "programming" tasks?

In order to test this two more objectives would be looked at:

- To provide tasks the successful completion of which would demonstrate the use of CT.
- To provide tasks that would demonstrate they can generalise what they learned to using other software.

The initial exposure of the large survey group would be done through a traditional class in UPSI. However, to answer RQ3 and 4 it would be necessary to conduct a more detailed instruction in order to spend time looking at the learner's behavior. Thus a second phase of the study was proposed to tackle these questions. This would involve a small group of students from a similar cohort to the previous phase being given one on one teaching in the use of Flash while being observed for their behaviour as well. The evaluation of this leads to a further research question and allows an automated teaching/learning tool to be designed focused on a user centred design approach.

RQ 5 Does the use of a suitable tool in learning create users who will be motivated to go on programming?

Objectives:

- To identify changes in attitude after use of the tool compared to before.
- To identify what barriers still remained.

Again positive outcomes from the previous research questions will lead to a final research question which asks about the possible implications of the study as a whole.

RQ 6 What are the implications of this study for the teaching of CT and programming in teacher training courses in Malaysia?

The final question can only be completely answered when the research is over and it is possible to engage with the authorities in Malaysia. At the time of writing the Malaysian government have already started to ask questions which this research will have begun to answer.

1.3 STRUCTURE OF THE THESIS

The remainder of this thesis is structured as follows:

Chapter 2 commences with a review of the relevant literature and outlines the broad perspectives and related concepts – programming and CT – which are then linked to the discussion of findings in chapter seven. The subjects of the literature review are designed to aid the answering of the research questions. So for example RQ 1 begins to be answered by knowing what the literature has to say about attitudes of teachers to learning programming.

Chapter 3 frames the methodological considerations in carrying out this investigation. It presents the methods that were used in this study, including recruitment and participant selection, and discusses the potential ethical concerns and pitfalls of the approach. The methods look particularly at ways of gathering data on the attitudes of the teachers and trainee teachers towards learning to teach programming. Thus they involve questionnaires and the use of open ended questions. Chapter 3 seeks to give background to why these research methods were appropriate.

Chapter 4 presents Study 1 which involves observations of Malaysian pre-service teachers at UPSI in order to identify difficulties experienced with learning to program. This will give further answer to RQ1 in particular and RQ2 in part.

Chapter 5 describes Study 2 which looks at how an individual one-on-one approach to teaching programming works among pre-service and in-service teachers. In doing so, it seeks to identify where non-CT exists and how it affects the learning of programming. It also seeks to identify ways in which to change that thinking. In doing so it lays the basis for the design of the teaching/learning tool that is presented in Chapter 6 and allows RQ2 and RQ 3 to begin to be answered.

Chapter 6 presents Study 3, the implementation of a tool to automate the teaching of CT skills and its evaluation with Malaysian pre-service teachers and in-service teachers who were resident as students in the UK. In doing so it looks at providing further answers to RQ 4 and RQ 5.

Chapter 7 discusses the research findings in light of the proposed research questions. It looks at how these findings contribute to the existing knowledge of how to teach computer programming, existing beliefs regarding the role of CT in this and the implications of this research. Chapter 7 also concludes in revisiting the research questions and summarizing the findings of this research. It discusses potential future work related to RQ 6 – addressing its uptake in practice in Malaysia for teaching teachers to be ready to teach programming and presents the specific contributions of this PhD. The expected overall contribution is detailed in Section 1.4.

1.4 EXPECTED CONTRIBUTIONS

Teaching non-programmers how to think computationally will be of immense value in education. It is expected that this work will demonstrate how to aid and even motivate the acquisition of CT skills and that this will offer the opportunity to challenge and change teachers' thinking. Ultimately, it is expected that if thinking regarding programming is changed then education will be more powerful in the field of technology and the consequential development of CT in users will additionally benefit other areas of life especially for Malaysian primary school teachers.