Journal of educational
and psychological sciences
Issue (9), Volume (2)
April 2018
ISSN: 2522-3399

المجلة العربية للعلوم و نشر الأبحاث
Arab Journal of Sciences & Research Publishing
AJSRP

Training Teachers in the Use of Programming and Computational Skills in the Classroom

Wedian Turki Mhmoud Ahmed Abuhussain

Western Michigan University || USA

Abstract: Schools have a responsibility to find ways to enable students to gain skills, such as sophisticated thinking, flexible problem solving, and communication skills. Computational thinking skills have been shown to be valuable for gaining the necessary skills to enter the 21st century workforce. STEM careers demand computational skills, but they are also necessary for all higher order thinking and computer-supported problem-solving careers. In general, teachers have left the teaching of computational skills to the technology or computer teachers at their schools. It has become apparent that all K-12 teachers must become adept at programming and computational thinking, and also they should be equipped to teach these skills to their students. The aim of this study was to identify current teacher knowledge of programming and computational thinking and develop their understanding of how it should be used in K-12 classrooms.

The researcher gathered quantitative data through an online survey to discover the current state of knowledge about computational thinking and its classroom use. Following the analysis of the data regarding teacher awareness, understanding, and use of computational thinking in their classrooms, the researcher developed recommendations for teacher education, including access to online materials to aid teachers in their use of computational thinking. Results suggested teachers would benefit from online training videos that could be integrated into teacher in service activities to instruct teachers in the use of coding and computational thinking in their science and math classroom instruction. Training teachers in programming and computational thinking has the benefit of making learning programming easier and fun, encourage teachers and students to learn programming, allow teachers to search and find other interesting programming websites, teach students how to program and play with code, and help students to improve their problem solving and thinking skills in a new and important way.

Keywords: Computational thinking; programming; k-12 education; teacher training.

Introduction

The concept of school success changes according to the skills needed in the 21st century. It is important for students to achieve mastery of these skills in order to be ready for university and work. That means students need to gain skills that allow them to keep up with the requirements of college and work environments. There are many challenges facing 21st century teachers, but the addition of technology to the classroom experience is the most influential factor.

It is obvious that technology plays an important role in helping students and schools achieve and meet 21st century skills, and schools are becoming quite adept at integrating technology into classrooms in order to help students gain these skills. This is not an easy challenge for schools and teachers to meet

because it is difficult to find currently-available, effective curriculum and tools. Schools have a responsibility to find ways that will enable students to gain skills, such as sophisticated thinking, flexible problem solving, and communication skills. They need these skills to succeed in work and life in the present day (Binkley et al., 2014).

All these 21st century skills are important, but this paper considers computational thinking because it is an essential skill, important in developing other higher-order thinking skills. Also, this paper will suggest that teachers must be trained to help their students acquire and practice computational thinking skills.

Wing (2006) defines computational thinking as a way of "solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science" (as cited in Barr, Harrison & Conery, 2011). Computational thinking is considered to be a fundamental skill because it is related to many aspects of 21st century skills, such as creativity, critical thinking, and problem solving (Ananiadou & Claro, 2009; Binkley et al., 2012, as cited in Lye & Koh, 2014, p. 52). Mishra and Yadav (2013) mention that computational thinking requires problem solving since this kind of thinking uses different strategies such as algorithms, abstraction and debugging. In addition, computational thinking can be applied in many disciplines.

According to Wing, students of all ages need to develop computational thinking skills because they need analytical ability in reading, writing, and arithmetic. "Thinking computationally or like a computer scientist can help students better articulate and comprehend other disciplines" (Kafai, Burke & Resnick, 2014, p. 6). When students learn computational thinking, they will develop and enhance many aspects of thinking skills, which help students to solve and address problems in an innovative way. They can apply the concept of computational thinking in many areas. For this reason, computational thinking can be considered a foundational and primary skill that all students need to learn (Einhorn, 2012; Mishra & Yadav, 2013).

According to Lye and Koh (2014), teaching programming to K-12 students is an effective way to teach computational thinking skills because programming utilizes all the components of computational thinking and the knowledge gained. Programming helps students to gain the experience of tackling programming challenges, which can provide a framework not only for computer science, but it also covers other fields from natural and health sciences to the social sciences and humanities (Einhorn, 2012). In fact, programming enables students to use computer science concepts, including abstraction, debugging, remixing, and iteration to solve problems. This way of thinking is considered a fundamental skill for K-12 students because they have to be "thinking at multiple abstractions" (Lye and Koh, 2014). As Steve Jobs said, "Everyone should learn to program a computer … because it teaches you how to think" (as cited in Kafai, Burke & Resnick, 2014, p.125). Integrating programming into K-12 classes is a powerful tool to improve students' computational thinking.

Research Materials

There has been a great deal of research seeking to explain the concept of computational thinking. This review will focus on computational thinking in education, especially as it applies to K-12 schools. Wing (2006) wrote an influential article on computational thinking in which she argued that adding this skill to every child's analytical ability was an essential addition to science, technology, engineering, and mathematics (STEM) learning (Grover & Pea, 2013). A report on computational thinking by the National Research Council (NRC) seems to agree that computational thinking is considered a cognitive skill which the "average person is expected to possess" (Yadav, Zhou, Mayfield, Hambrusch & Korb, 2011, p.1).

Computational thinking requires using digital computers to enhance problem solving and thought processes. It helps to enhance, facilitate, and expand the realm of thinking, logic, and mathematics. Scientists and researchers in different fields such as biology, chemistry, physics, medicine, and others, actually need to use digital computers for "mathematical modeling" to develop the frontiers of knowledge. Thus, it is clear that our "knowledge-based" world requires all of us to realize how to utilize computational thinking in everyday activities in order to increase productivity in the workplace (Sanford & Naidu, 2016).

Furthermore, computational thinking leads to the use of scientific methodologies, and improves both inventiveness and innovative thinking. It is a core skill in mathematics, engineering, technology, and science. This way of thinking is the starting point to generate enormous changes and benefits in these fields (Einhorn, 2012). Similarly, Hemmendinger (2010) points out that the goal of teaching computation thinking "is to teach them how to think like an economist, a physicist, an artist, and to understand how to use computation to solve their problems, to create, and to discover new questions that can fruitfully be explored" (as cited Yadav, Zhou, Mayfield, Hambrusch & Korb, 2014, p.2).

In the current labor market, jobs require solving unstructured problems, communication, and non-routine manual work. Under these circumstances, young people have to be prepared to adapt. Two main types of tasks that depend on uniquely human abilities are (1) integrating many types of information to solve unstructured problems and (2) acquiring, making sense of, and communicating information to others. That means computational thinking is a key skill in the realm of solving problems, interpreting and understanding data, and communicating information with others by computers (Lee, Martin & Apone, 2014). Moreover, there are many attitudes or dispositions that are the result of computational thinking skills, such as:

- Confidence in dealing with complexity.
- Persistence in working with difficult problems.
- Tolerance for ambiguity.
- The ability to deal with open-ended problems.

- The ability to communicate and work with others to achieve a common goal or solution (Barr, Harrison & Conery, 2011, p.21).

According to Angeli et al. (2016), "computational thinking is a thought process that utilizes five skills: abstraction, generalization, decomposition, algorithmic thinking, and debugging" (p.49). Angeli and associates also provide a computational thinking curriculum framework that shows indicators of efficiency for all of the beforementioned five computational thinking skills, in a progression from simple to complex across the educational levels of K-2, 3-4, and 5-6.

The goal of this framework is to engage students in thinking and problem solving by suggesting a solution to a problem, automating the solution through algorithmic thinking, and generalizing this solution to new problems when common patterns are determined (Angeli et al., 2016). The positive sign of integrating computational thinking into classrooms is that students have considerable capability to comprehend computing concepts, which are considered as the main components of teaching computational thinking in other disciplines (Lewandowski et al., 2007, as cited in Yadav et al, 2014).

The study by Lewandowski and his colleagues (2007) emphasized that students from different backgrounds used analytical and problem-solving skills to solve a concurrency task in a beginning CS1 class (that is the first course in the introductory sequence of a computer science major, and typically focuses on problem-solving techniques and computer programming). The researchers found that 69 percent of the answers were correct, which meant students from different backgrounds area were already able to understand basic concepts to solve computing problems (as cited in Yadav et al, 2014). This proves that integrating computational thinking can effectively start in K-12 education and can teach students how to utilize the computational thinking skills.

The obvious question is what way can technology be integrated into K-12 education in order to improve computational thinking for students? The useful way to learn computational thinking is programming. In fact, programming is considered as the broader and more multifaceted process, so it can be seen as a tool for developing all CT aspects (Mannila et al, 2014). In addition, many studies suggest that making programming a more common skill for all, and introducing algorithms ('rithm) as the fourth "r" (the Three Rs are: reading, writing, and arithmetic) for 21st-century literacy (Grover & Pea, 2013).

However, "in order to make it possible to use programming as a tool for addressing all CT capabilities, there needs to be some common ground and consensus with regard to what programming is and how it should be introduced at different levels taking into account the age of the students" (Mannila et al, 2014, p.17). Students are more comfortable learning programming through visual presentations such as diagrams, videos, and animation. Furthermore, when students use drag and drop programming applications (blockly coding), they develop higher order thinking skills. Many programming platforms have evolved in terms of blockly coding such as Scratch, Alice, Blockly, and Kodu. Code.org is one of

blockly coding online platforms used to teach student programming by using a visual programming language (Kalelioglu, 2015).

The first step in integrating computational thinking in teaching is to prepare K-12 teachers to present computational thinking ideas in explicit ways. There are many suggestions about how to do this. One suggestion is that "students could learn about computational thinking by observing teachers as they model related thinking strategies and guide students to use these strategies independently" (Yadav et al., 2014, p.3). However, generally, the efforts to educate teachers about computational thinking focus on computer science teachers. For example, Blum and Cortina held workshops to introduce computational thinking to high school computer science teachers. These workshops were aimed to raise teachers' awareness and provide materials that they could use to show students that computer science was more than just programming. As a result, these workshops illustrated teachers' improved understanding of computational thinking and increased awareness of the importance of computational thinking in all aspects of life (2007, as cited in Yadav et al, 2014).

However, to achieve the purpose of integrating computational thinking, all teachers, and not just computer science teachers, have to become involved in the process. One way to reach this goal is to begin at the preservice teacher level and introduce computational thinking concepts in teacher preparation programs. Preparing preservice teachers in computational thinking concepts might allow them to see the relevance of computational thinking in their own disciplines (Yadav et al, 2014). The key for successful incorporation of computational thinking into the K-12 curriculum is to prepare future teachers to teach it (Yadav et al., 2011).

For example, Israel and his colleagues (2015) did a study that aims to investigate how elementary school teachers with limited computer science experience in a high-need school integrated computational thinking into their instruction. The study began with a week-long summer workshop in which twenty of the teachers in the school volunteered to participate. During this workshop, they were introduced to the basics of computing and computational thinking, introduced to modeling of computing tools, and were given time to practice using these computing tools. The next workshop, during the winter break, included additional professional development and allowed the teachers time to work on integrating computing into their instruction.

It becomes apparent from looking at the literature that teacher education on programming skills is one of the key components to integrating computational thinking into K-12 education.

Problem Statement

It is clear that computational thinking involves many 21st century skills and capabilities. Schools are searching for ways to integrate computational thinking into their curriculum in order to help their students gain computational thinking skills. Many schools find that integrating programming is an

effective way to improve computational thinking, especially because there are now many online free sources such as Code.org, Scratch, and Cargo-Bot, all available for classroom use.

Teachers who have been previously trained in teaching methods do not generally have the necessary skills to integrate programming into their K-12 classrooms as a tool for teaching computational skills. Additionally, textbooks that are more than five or ten years old generally do not include programming concepts, and teachers find it difficult to gain programming skills for their own use as well as for the use of their students.

The problem becomes one of finding ways to train teachers in the use of programming and show teachers how to teach programming in a fun and easy way. Unless they have been trained in programming, most teachers do not know what they are missing.

Methods

The methodology used in this study was two-fold. The first was to discover the current state of knowledge about computational thinking and how it could be used in the classroom. The researcher created a needs analysis based on quantitative data gathered in an online survey. The online survey has many advantages because it can access a group of people in different areas and save time and cost. Google Forms was used to create the online survey. (Google Forms provides a free survey with unlimited respondents.) A random sample of K-12 teachers, administrators, and technologists in Saudi Arabia and the United States was created. In order to capture the sample, the survey was distributed by a link to 100 teachers, and 78 forms were received with completed responses. The sample includes 60 percent K-12 teachers, 23.75 percent K-12 administrators, and 16.25 percent K-12 technologists. The sample covers all K-12 levels: 38.5 percent Primary, 39.7 percent Middle, and 21.6 percent Secondary. The survey was analyzed and results propelled the next phase of the study.

The second methodology analyzed teacher awareness, understanding, and use of computational thinking in their classrooms. Following the analysis, a protocol was developed to provide suggestions and recommendations for teacher education for both preservice teachers and K-12 educators. Classroom and in service education materials were developed and recommendations were made to schools and university educators to aid in the development and understanding of computational thinking among educators. The major recommendation was to provide access to online materials to aid teachers in their use of computational thinking. A website, the Hour of Code, and set of online lesson plans helped teachers obtain the skills necessary to integrate computational skills into their classroom teaching. The website included videos about computational thinking and what it means for 21st century education, what the benefits of integrating computational thinking into classrooms are and it gave some examples of tools that could be used for both teaching and learning. Additionally, the website provided a list of tools and resources about computational thinking that teachers could use as a reference. The information is free and

available online at code.org for any schools or teachers who are interested in teaching computational thinking for students. The videos and website provided a content that meets the teachers' needs to improve ways of integrating computational thinking into teaching.

Results

Based on the information gathered from the random sample, the following table shows teachers' level of knowledge about computational thinking as it was relayed through the surveys.

Table 1: Teachers' Level of Computational Thinking Knowledge

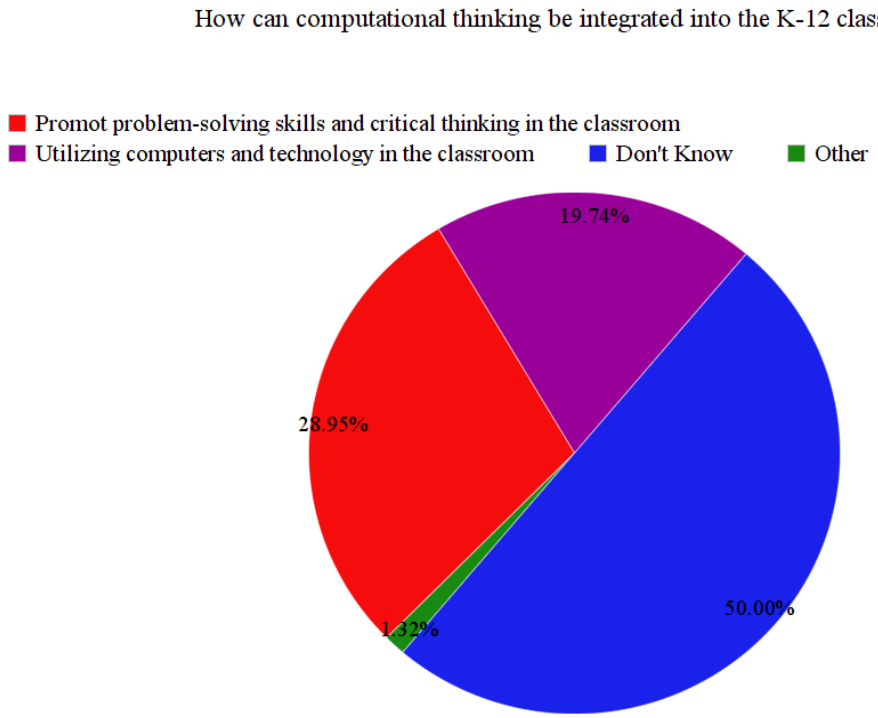
Question	Not at all knowledgeable	Somewhat knowledgeable	Knowledgeable	Very knowledgeable
What is Computational Thinking?	53.3%	24%	18.6%	4%
How Computational Thinking connects to what you are already doing in your classroom?	55.5%	29.7%	9.7%	5.5%
Which concepts in your curriculum lend themselves to being taught using Computational Thinking tools and approaches?	59.7%	23.6%	11.1%	5.5%
Which concepts in your curriculum would benefit from being taught using Computational Thinking tools and approaches?	56.9%	25%	11.1%	6.9%

As can be seen from the table, more than fifty percent of the sample did not have any knowledge about computational thinking nor any understanding about how to integrate computational thinking into their classrooms. Additionally, surveys indicated that most teachers did not have any experience with computer programming.

When asked about integrating computational thinking into K-12 classrooms, fifty percent of teachers responded "do not know." The teachers were offered basic options such as promote problem-solving skills and critical thinking in the classrooms and utilize computers and technology in the

classroom, but, as can be seen from the following chart, they still communicated that they did not know what to do.

Integrating computational thinking in the K-12 classroom



It became obvious to the researcher that there was a distinct need for K-12 teacher training in computational thinking and computational thinking skills. Therefore, recommendations were made to schools that comprehensive training should be developed to help teachers integrate programming into their classrooms.

The surveys indicated that teachers could benefit from online training videos on a website. The website could be created and available for any school seeking to improve its students' computational thinking. The main purpose of this website is to introduce teachers to the benefits of teaching programming and how to encourage their students to learn programming. Training would also aim to encourage schools to allocate one to two hours per week for students to practice programming. That will helps students to develop computational thinking, which would directly enhances other thinking skills used in curriculums, such as Math and Science.

In addition, teachers will be taught how to use the "Hour of Code" website in order to teach students programming. The Hour of Code is a website that is organized by code.org, which helps students to learn how to program with online tutorials that are available for all ages in over 45 languages.

According to Kalelioglu (2015), "code.org presents the opportunity for students to learn computer science through drag and drop programming. The site is structured as a game, and consists of self-directed tutorials, posters and video lectures by famous people such as Bill Gates and Mark Zuckerberg as a way to help inspire students to look at computer sciences" (p.202). The most significant feature of the website is that it is an effective, easy, and fun way to teach programming. In the creation of such a training program, three videos would be developed for teachers. One would provide an introduction and general ideas regarding the concept of computational thinking. The other two videos would be tutorials designed to show teachers how to use the Hour of Code website.

Discussion

Computational thinking is considered one of the essential skills for students in the present day. Computational thinking helps students to master many subjects such as math, technology, and science. STEM (Science, technology, engineering and mathematics) are the disciplines deemed most necessary for the 21st century, and requires a systemic change that includes teacher engagement, student education, and the development of significant resources. So, schools must do their utmost in order to develop students' computational thinking skills, and teachers play the most important role in this process.

Results of the study showed that teachers are ill-prepared to utilize computational thinking into their classrooms. First, teacher education need to develop curriculum that will teach programming and computational thinking skills to pre-service educators. Additionally, school districts can and should develop materials that will educate teachers already in service in programming and computational thinking. From the results of this paper's needs analysis, a large percentage of teachers do not know what computational thinking means, nor do they understand how to integrate computational thinking into their teaching. In addition, teachers need to know and understand what tools they can use to teach computational thinking.

Yadav et al.(2014) suggested that the way to provide teachers with the necessary skills is to begin at the preservice teacher level and introduce concepts in teacher education programs. This presupposes, of course, that technology will be available for preservice teachers as well as practicing educators. Fundamental data organization methods taught in these classes can help preservice teachers understand how computational thinking works and remind them of how they use computational thinking in their own lives. The second step addresses applications for teaching and learning, including abstraction, reasoning skills, and patterns in data. In pedagogy and curriculum classwork, preservice teachers can gain practice in designing lesson plans that utilize computational thinking in their content.

For practicing educators, particularly for those who have taught several years without technology, the training will need to be conducted through in service activities. A school district might work in conjunction with computer science educators, science, and social science educators to create a series of

workshops that would help practicing educators understand the value of computational thinking in their classrooms. An introduction through the Hour of Code website, as an example, would definitely help create in educators a deeper understanding of coding and computational thinking.

Training teachers in programming and computational thinking will have the benefit of making learning programming easier and fun, encourage teachers and students to learn programming, allow teachers to search and find other interesting programming websites, teach students how to program and play with code, and help students to improve their problem solving and thinking skills in a new and important way.

Conclusion and Recommendations

The aim of the current study was to make educators aware of the necessity of incorporating computational thinking into the pedagogy of elementary and secondary education. To that end, a survey was developed to assess the current knowledge among educators of programming and computational thinking. One hundred surveys were sent out, and 78 were submitted. It was discovered that most currently practicing educators had little understanding of programming or computational thinking, and most had no knowledge of how to implement computational thinking into their curriculum. To aid teachers in their development, in service materials were developed to encourage teachers to create curriculum that would help their students understand and utilize computational thinking.

Schools have a responsibility to find ways to enable students to gain skills, such as sophisticated thinking, flexible problem solving, and communication skills. Computational thinking skills have been shown to be valuable for gaining the necessary skills to enter the 21st century workforce. STEM careers demand computational skills, but they are also necessary for all higher order thinking and computer-supported problem-solving careers.

The survey conducted for this study indicated that K-12 teachers are generally ill-equipped to understand higher-order computational skills, let alone teach them to their students. They have usually left the teaching of computational skills to the technology or computer teachers at their schools. However, trends in STEM education, as well as social studies indicate that code and computational thinking have become essential components of K-12 education. It is now essential for K-12 teachers to become adept at programming and computational thinking and also be able to teach these skills to their students.

The results of the study brought forth several suggestions for further training for educators to learn how to adopt computational thinking methods into their classrooms. It was suggested that training sessions, and perhaps training videos, could be developed for school districts and regional educational services to help teachers understand computational thinking and why and how they adapt classroom curriculum and pedagogy. It is recommended that basic programming should be taught through self-learning, school-developed training sessions, and in-service education. Every entity should design their

own program to meet the growing need for computational thinking. The Hour of Code website is just one example of the type of training that could be offered to K-12 teachers.

Twenty-first century students will spend much of their time utilizing technology in their education and in their careers. It is essential that teachers are prepared to meet those needs and create the understandings necessary to produce a competent future workforce.

References

1. Angeli, C., Voogt, J., Fluck, A., Webb, M., Cox, M., Malyn-Smith, J., & Zagami, J. (2016). A K-6 computational thinking curriculum framework: Implications for teacher knowledge. Journal of Educational Technology & Society, 19(3), 47-57. Retrieved from: http://libproxy.library.wmich.edu/login?url=http://search.proquest.com.libproxy.library.wmich.edu/docview/1814440935?accountid=15099
2. Barr, D., Harrison, J., & Conery, L. (2011). Computational thinking: A digital age skill for everyone. Learning & Leading with Technology, 38(6), 20-23.
3. Binkley, M., Erstad, O., Herman, J., Raizen, S., Ripley, M., Rumble, M., & Miller-Ricci, M. (2014). Defining twenty-first century skills. In Assessment and Teaching of 21st Century Skills (pp. 17-66). Springer Netherlands.
4. Einhorn, S. (2012). Microworlds, computational thinking, and 21st century learning. LCSI White Paper.
5. Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38-43.
6. Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. Computers & Education, 82, 263-279.
7. Mishra, P., Yadav, A., & Deep-Play Research Group. (2013). Rethinking technology & creativity in the 21st century. TechTrends, 57(3), 10-14.
8. Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K--8 curriculum. ACM Inroads, 5(4), 64-71.
9. Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12?. Computers in Human Behavior, 41, 51-61.
10. Kafai, Y. B., Burke, Q., & Resnick, M. (2014). Connected code: Why children need to learn programming. MIT Press.
11. Kalelioglu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. Computers in Human Behavior, 52, 200.

12. Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014, June). Computational thinking in K-9 education. In Proceedings of the working group reports of the 2014 on innovation & technology in computer science education conference (pp. 1-29). ACM.
13. Sanford, J. F., & Naidu, J. T. (2016). Computational thinking concepts for grade school. Contemporary Issues in Education Research (Online), 9(1), 23.
14. Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011, March). Introducing computational thinking in education courses. In Proceedings of the 42nd ACM technical symposium on computer science education (pp. 465-470). ACM.
15. Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., & Korb, J. (2014). Computational thinking in elementary and secondary teacher education. ACM Transactions on Computing Education (TOCE), 14(1), 1-16.

تدريب المعلمين من الروضة الى الصف الثاني عشر لاستخدام التفكير الحسابي في الفصول الدراسية.

الملخص: تتحمل المدارس مسؤولية إيجاد طرق لتمكين الطلاب من اكتساب المهارات، مثل التفكير المتطور، وحل المشكلات المرنة، ومهارات الاتصال. وقد تبين أن مهارات التفكير الحاسوبي تكون ذات قيمة لاكتساب المهارات اللازمة لدخول القوى العاملة في القرن ال 21. وظائف ستيم تتطلب المهارات الحسابية، ولكنها ضرورية أيضا لجميع الأنظمة العالية التفكير والحلول التي تدعمها الكمبيوتر حل المشاكل. بشكل عام، ترك المعلمون تعليم المهارات الحسابية لمعلمي التكنولوجيا أو الكمبيوتر في مدارسهم. وقد أصبح واضحا أن جميع المعلمين من الروضة حتى الصف الثاني عشر يجب أن يكونوا بارعين في البرمجة والتفكير الحسابي، وأيضا يجب أن تكون مجهزة لتعليم هذه المهارات لطلابهم. وكان الهدف من هذه الدراسة هو التعرف على المعرفة الحالية للمعلمين من البرمجة والتفكير الحسابي وتطوير فهمهم لكيفية استخدامه في الفصول الدراسية.K-12

جمع الباحث البيانات الكمية من خلال استطلاع على الانترنت لاكتشاف الحالة الراهنة للمعرفة حول التفكير الحسابي واستخدامه في الفصول الدراسية. وبعد تحليل البيانات المتعلقة بتوعية المعلمين وفهمهم واستخدامهم للفكر الحسابي في فصولهم الدراسية، قام الباحث بوضع توصيات لتعليم المعلمين، بما في ذلك الوصول إلى المواد الإلكترونية لمساعدة المعلمين على استخدام التفكير الحسابي. وقد اقترحت النتائج أن يستفيد المعلمون من أشرطة الفيديو التدريبية عبر الإنترنت التي يمكن دمجها في أنشطة تقديم المساعدة للمعلمين لإرشاد المعلمين في استخدام الترميز والتفكير الحسابي في تعليمهم في الصفوف العلمية والرياضية. تدريب المعلمين في البرمجة والتفكير الحسابي لديه فائدة لجعل برامج التعلم أسهل وممتعة، وتشجيع المعلمين والطلاب على تعلم البرمجة، والسماح للمعلمين للبحث والعثور على غيرها من مواقع البرمجة للاهتمام، وتعليم الطلاب كيفية البرمجة واللعب مع التعليمات البرمجية، ومساعدة الطلاب لتحسين مهاراتهم في حل المشاكل والتفكير بطريقة جديدة وهامة.

الكلمات المفتاحية: التفكير الحسابي. برمجة التعليم من الروضة الى الصف الثاني عشر. تدريب المعلمين