



The effect of computational thinking skill program design developed according to interest driven creator theory on prospective teachers

Ozlem Uzumcu¹  · Erdal Bay²

Received: 16 January 2020 / Accepted: 30 June 2020 / Published online: 12 July 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

It is likely to observe that the increase in data and the interrelated challenges in digital age complicates the problems to be encountered. Therefore, unprecedented problem-solving skills have become inevitable. Though dating back to old times, computational thinking skill is defined as a recent skill area that is required by everybody, that can be used to solve the aforementioned complex problems, and that is included in international standards and training programs. In this study, it was aimed to improve computational thinking skills of prospective teachers. In order to do this, a program design which includes contents that prospective teachers can use in daily life and professional life has been developed. This program, which consists mostly of unplugged activities, also includes computer aided and robotic activities. A total of 11 voluntary prospective teachers (7 women and 4 men), who were in their 3rd year of the 4 year education in the 2017–2018 fall semester and did not attend to any programming or computational thinking education training before, participated in the study. In the first application, a 40-h program was carried out with five prospective teachers, while in the second application, an updated 52-h program was carried out with six prospective teachers. A skill test was developed, and applied to measure prospective teachers' computational thinking skills before and after the prepared program. Moreover, at the end of the training, they were asked to prepare graduation projects and their perspectives on education were examined. It has been observed that the program applied to prospective primary education teachers, who did not take any lessons like programming etc. before, was effective according to the computational thinking related skill tests and their graduation projects. It has been also observed in prospective primary education teachers that their thinking skills such as problem solving and questioning were improved and they could reflect their acquired knowledge and skills to their daily and professional life.

Keywords Computational thinking · Twenty-first century skill · IDC theory · Unplugged activities · Program design

✉ Ozlem Uzumcu
ozlem.uzumcu@hku.edu.tr

1 Introduction

Today, it is possible to observe that skill-oriented training is needed to meet the requirements of the era. At this point, different institutions and organizations have explained what these skills are, their scope and definitions, in order to gain twenty-first century skills from a young age. These skills have been found to include areas such as problem solving, critical thinking, social skills, technology literacy (ISTE 2011; OECD 2018; Partnership for 21st Century Skills 2015). Problem solving skills are among the twenty-first century skills and are among the basic skills needed from past to present. (Kotluk and Kocakaya 2015; Voogt and Roblin 2010).

Problem solving skill is defined to be a complex process involving cognitive, affective and behavioural skills (Korkut 2002), and it is described to be solving a complex problem or a difficulty encountered based on another definition (Özsoy 2014). It can be claimed that the reasons of complexity such as the rapid increase in information sources, the rise of communication channels through these sources and the effect of technology on all those sources are reflected in today's problems. In this regard, computational thinking emerges as the new twenty-first century skill to be used in solving complex problems. Although computational thinking, which is identified to be a kind of problem solving skill in its most general sense, dates back to old times (Papert 1980), not much has been mentioned about its precise content frame (Grover and Pea 2013; Hu 2011).

The headings of decomposition, abstraction, pattern, algorithm, evaluating and debugging can be found directly or indirectly in the content of computational thinking (ISTE 2018; CSTA 2017; Google Education 2017; LEGO Education 2018a, 2018b; Brennan and Resnick 2012; K-12 Computer Science Framework 2016; Kalelioğlu et al. 2016; Wing 2008). The dimensions of computational thinking are explained as in Fig. 1, based on these sources. These concepts are briefly mentioned as follows.

- *Decomposition*, which is characterized to be the disassembling of the problem, is usually the first dimension confronted in the sources for an easier solution of the problem (Wing 2006; Google Education 2017; LEGO Education 2018a, 2018b). It is stated that while *abstraction* which enables the review of the relationships between the problems in the solution processes can be the most difficult dimension of computational thinking (Booth 2013), it constitutes the essence of computational thinking (Wing 2008).
- Abstraction which is regarded to be the basic condition of problem solving is also interpreted as the process of generalization from certain examples (Wing 2011; Lee et al. 2011).
- It is ascertained that expressions such as *pattern*, pattern recognition, pattern / model extraction are used in computational thinking. Its definition by Google Education includes observing patterns, trends and layouts in data (Google Education 2017).
- *Evaluating and debugging* can be defined as the process of evaluating the steps taken to check the accuracy or the elimination of existing errors (Weiser 1982). It is known that the procedure of debugging requires an understanding beyond the ability of writing codes, and therefore it is significant in the process of programming (Liu et al. 2017).

- Another computational thinking heading appears to be *algorithms* that are the bases of programming. Algorithm is defined as performing a task or job step by step in any area (Selby and Woollard 2013). Computational thinking, which is claimed to affect research in almost all the disciplines of science, is also labelled as algorithmic thinking (Bundy 2007; Barr and Stephenson 2011). While it is asserted that there are points where the computational thinking intersects with the algorithmic thinking, there are also studies implying that computational thinking is based on algorithmic thinking (Denning 2009; Lee et al. 2011). It is also expressed that computational thinking requires more skills than programming (Lu and Fletcher 2009). The indispensable relationship between computational thinking and programming (Israel et al. 2015) explains the popularization of programming teaching and computational thinking skills in the same period.

In this process, computational thinking is becoming widespread in a wide range from curriculums to international standards (Qualls and Sherrell 2010). Accordingly, while computational thinking is one of the ISTE student standards in 2016, the skill is addressed with its own standards in 2019 (ISTE 2016; ISTE 2019). Moreover, it is

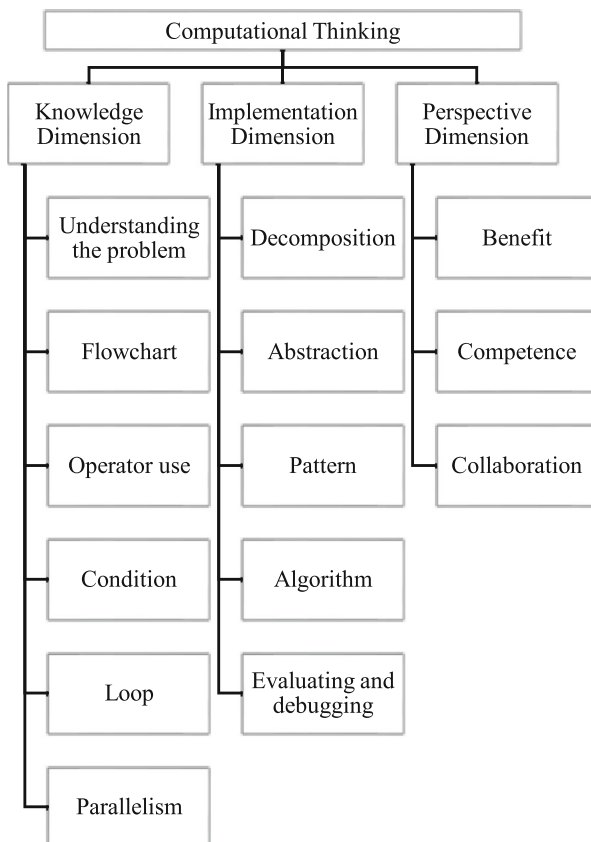


Fig. 1 Dimensions of computational thinking

witnessed that computational thinking and its dimensions are widely used in international computer frameworks (CSTA 2017; K12.org, 2018). All these developments can be considered to be the herald that computational thinking will be the subject of further study.

When the studies related to computational thinking are examined, it is clear that they are mostly in the field of information technologies, and block based interventions and robotic kits are used to gain that skill to a great extent (Kalelioğlu et al. 2016). Although it is not a computer or technology-specific skill (Wing 2006, 2011), there are not many studies on the use of computational thinking in daily life or branch courses. Therefore, in this study, an innovative program design was needed to be able to use computational thinking skills in daily and professional lives and to integrate them with courses other than information technologies. While the use of unplugged activities in the majority provides accessibility to gain computational thinking skills, the inclusion of computer-aided and robotic activities will increase its variety of use (Curzon et al. 2014). It is indicated that problem solving skills can be transferred to non-programming courses thanks to the implementation and perspective dimensions of computational thinking (Lye and Koh 2014; Barr and Stephenson 2011; Resnick et al. 2009). The preparation of content specific to elementary teaching including many branches such as mathematics, science and technology according to Interest Driven Creator (IDC) theory also makes the study important.

For this purpose, a program was designed prospective teachers to use their computational thinking skills in their daily and professional lives. The effect of this program design, which consisted mostly unplugged activities and was based on the IDC Theory and formed with the information, application and perspective dimensions of computational thinking, was aimed to be examined. In this context, answers to the following questions were sought:

1. According to the analysis of pre-test and post-test results of the skills test related to the “knowledge and implementation” dimension of computational thinking based on i) Analytical rubric, ii) SOLO taxonomic analysis, how does the learners’ level of computational thinking skill change?
2. According to the analysis of graduation projects, which enable the prospective teachers to reflect their knowledge and skills as the graduation requirement of Computational Thinking Training, related to the i) knowledge, ii) implementation dimensions based on analytical rubric, what is the respondents’ level of computational thinking skills?
3. What are the opinions of the participants regarding the sub-dimensions of benefit, competence and collaboration of the perspective dimension of computational thinking?

2 Methodology

In this study, the design-based approach was preferred as an innovative educational intervention related to computational thinking, which is among twenty-first century skills, was attempted to be developed and its effects were investigated (Kuzu et al.

2011; Design-Based Research Collective 2003). This approach, known as design and development research, involves the processes of designing, developing and evaluating innovative products and practices (Richey and Klein 2014).

In this study, it is aimed primary education teachers to gain computational thinking skills that they can use in daily and professional life. A new program design was needed because of 3 reasons: the program to be designed should have content specific to the interests and needs of the target group, according to the IDC theory, majority of the activities being unplugged activities, and computational thinking being a new concept in Turkey.

Design based research, experimental research and action research were discussed in determining the method of the research. While experimental studies aim to test the hypothesis, product development is aimed in design-based research. Another reason why the experimental method is not preferred is that while variables are required to be under control in this method, all variables that may affect the design in the design-based method are included in the research. Action research, also known as teacher research, is similar to design-based research in some ways, but at some points it diverges. The most important difference between these two research type is that in action research it is aimed to increase the effectiveness of existing applications by eliminating the problems of them, while in design based research it is aimed to produce innovative theory, learning environment or educational practices in design based research (Design-Based Research Collective 2003; Kuzu et al. 2011). For these reasons, it was thought that the most appropriate method to design the program in this research was the design-based method. This method was preferred in this study, as it aims to promote innovation in education and to establish a stronger link between educational research and the real world (Bell 2004; Kuzu et al. 2011).

In this study, in accordance with the design-based method, program design related to computational thinking was developed with backward design, and it was applied to two groups. As seen in Fig. 2, after the program was applied to the first study group, the program was evaluated and necessary arrangements were made. The updated program was applied to the second group and re-evaluated and the program was finalized (application details are given in Table 1).

The study was carried out in a foundation university in Turkey where the participants study. The content of the training, which was conducted independently of the undergraduate courses, was briefly mentioned to the prospective teachers. The study involved 11 voluntary prospective teachers, seven females and four males, who were the juniors studying at the Department of Primary Education in the fall semester of 2017–2018 academic year and had no prior programming or computational thinking training.

2.1 Interventions

Backward design was preferred in the program design process. Accordingly, while determining the targeted results as given in Fig. 3, the content is obtained by determining the national and international standards and the expectations regarding the curriculum (Wiggins and McTighe 1998). Here, the mentioned is not the results of the study, but can be defined as the targeted learning outcomes. In this step, ISTE, which is international standards, CSTA, and K12 computer science framework (k12cs.

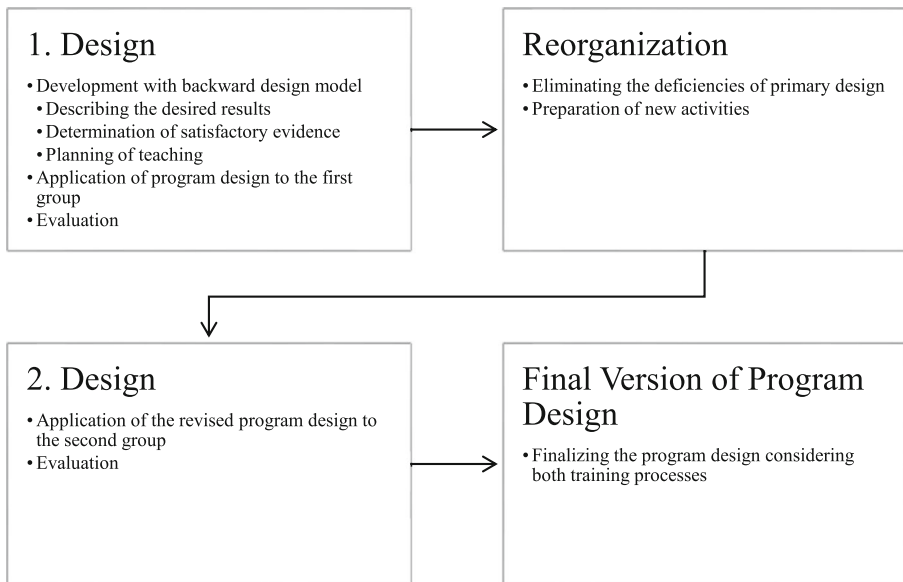


Fig. 2 The process of developing the program design

org) which includes international education programs were used, while determining the desired results.

The “determine acceptable evidence” section, which is the second step of the program design, can be considered as the determination of the evidence indicating whether the students have achieved the targeted results, determination and development of assessment and evaluation tools. Details of this step is given in the section “2.3. *Data collection instruments, processes and analysis*” of this article. The third step of the program design is “Plan learning experiences and instruction” section. This section, which contained planning of all the details about the instruction, was designed based on the IDC - Interest Driven Creator Theory. According to this theory, preparation of learning activities with content that appeals to the areas of interest for students is effective in permanent learning, because motivation which will provide innovative, compelling meaningful learning is needed in young people (Kong 2016). According to the same theory, individuals/students can be lifelong learners and creative thinkers if they participate in activities that interest them (Chan et al.,2015). The activities designed according to IDC not only help the students strengthen their computational thinking skills and perspectives but also provide an interesting theoretical basis in K-12 curricula (Kong 2016). It is suggested that the content that will be of interest to the students should be composed of current, authentic and daily life problems. Table 1 provides a summary of the program and the details of its implementation.

While the constructivist approach has been taken into consideration for the activities in the program, method and techniques such as discovery learning strategy, brainstorming and Socrates discussion in which the students are active have been preferred. In addition, collaborative scenarios have been created and peer education has been promoted through group activities in most of the activities. While unplugged activities made up the majority of program design, computer-aided and robotic activities were

Table 1 The content of the program directed towards computational thinking

Program Design	Approach Method Techniques	Activities	Period*	Number of activities by topic **
Dimensions of computational thinking	Constructivist approach Discovery learning strategy	39 Unplugged 3 Plugged	Weekdays 4.5 h on average for 9 days Total 40 h	Daily life 22 Mathematics 20 Technology 13 Logic 3
◦ Knowledge	Collaborative works Case study	4 Robotic-aided		Subject matter knowledge 4 Science 2
◦ Implementation	Brainstorming Drama			Professional life 2 Dance-Rhythm 1
◦ Perspective	Socrates discussion Group discussion Role playing	56 Unplugged 3 Plugged 4 Robotic-aided	Weekends 6.5 h on average for 8 days Total 52 h	Daily life 30 Mathematics 22 Technology 20 Logic 5 Subject matter knowledge 4 Science 3 Professional life 3 Dance-Rhythm 1

* Activities were designed according to IDC theory

**Some activities covered more than one subject area

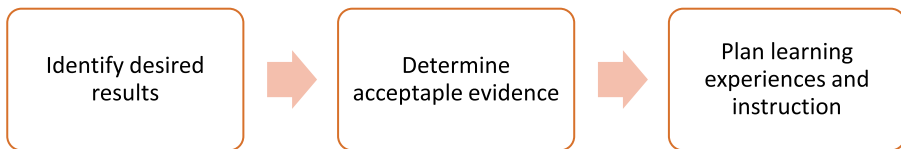


Fig. 3 Stages in the Backward Design Process Wiggins & McTighe, (1998)

also included to give insight into the different uses of computational thinking. The first application was carried out during 40 h on weekdays for 2 weeks, and the second one was conducted during 52 h at weekends for 4 weeks. Program design was reorganized, new activities were designed and some activities were updated according to the deficiencies after the first application, and the results of skill test and graduation projects.

2.2 Data collection instruments, processes and analysis

To determine the effects of program design, which was developed and applied on the prospective teachers, i) skill test ii) graduation projects and iii) perspective form (vv) were used. Data collection instruments and data collection and analysis are shown below (Fig. 4).

2.2.1 Skill test

While developing the skill test, the dimensions of computational thinking were considered and the usage of these dimensions in international standards (CSTA, ISTE, K-12 etc.) and the skills they measure in the curriculum were taken into consideration. Accordingly, it was determined that unplugged questions could contain more than one dimension (Fig. 5), and question types similar to visual logic questions could be included. For the skill test questions arranged in five items, two separate pilot applications were made and at the end of each

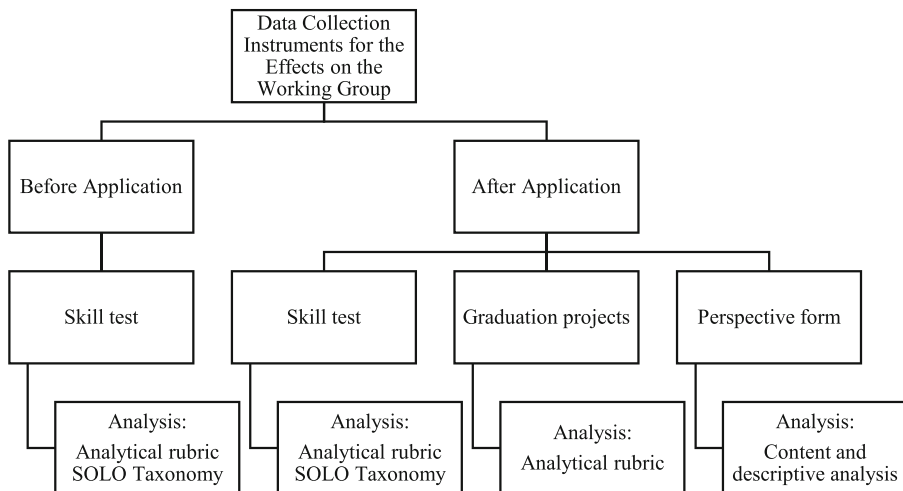


Fig. 4 Data collection instruments

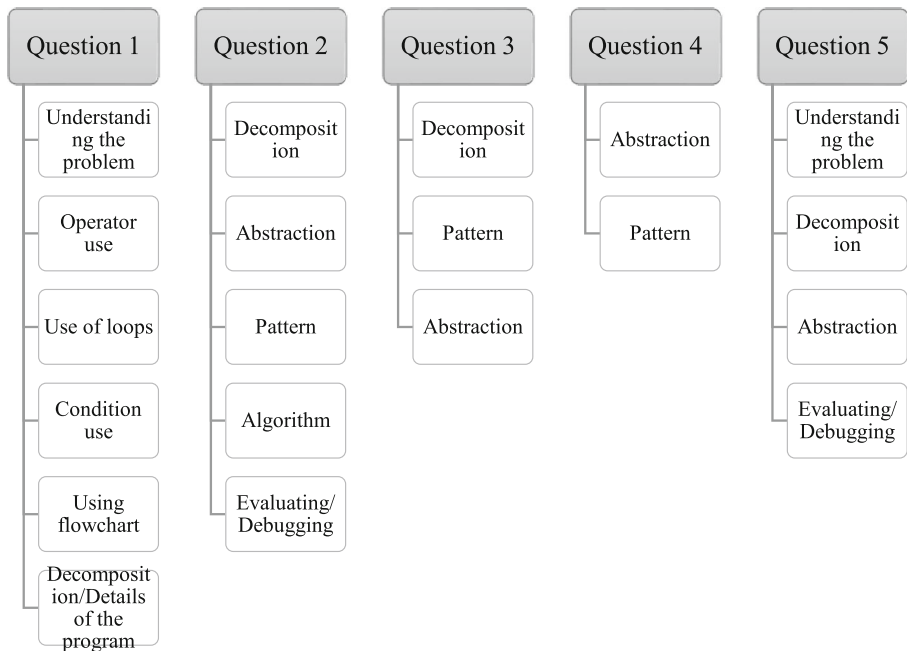


Fig. 5 Dimensions of computational thinking skill test questions

application, the skill test and its answer key were finalized following the expert opinion.

Written skill test was applied face-to-face within 50–60 min before and after the training. In the analysis of skill test, two different methods were followed: analytical rubric and solo taxonomy.

2.2.2 Graduation projects

Participants were asked to prepare graduation projects including i) knowledge ii) implementation dimensions related to computational thinking in order to complete the training. Graduation projects have been proposed as one of the appropriate ways to measure computational thinking skills (Brennan and Resnick 2012). Therefore, it has been explained to the prospective teachers that preparing the graduation projects is the requirement to complete the relevant education program and the necessary features of the project are explained. In Fig. 6, the essential features of the graduation project relevant to implementation step are given.

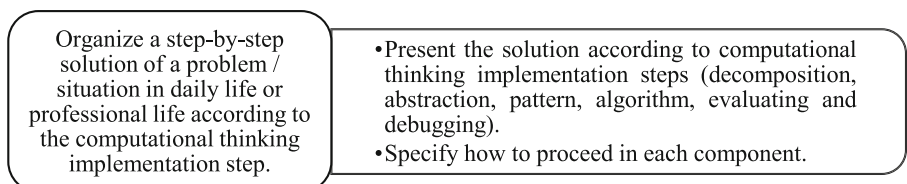


Fig. 6 Details of the graduation project requested from prospective teachers at the end of the training

While the analytical rubric of the knowledge dimension project consists of five headings (understanding the problem, operator use, use of loops, logical testing, using flowchart or writing as pseudo codes), the analytical rubric of the implementation step includes six headings (being a suitable example for computational thinking, decomposition, abstraction, pattern, algorithm, evaluating / debugging).

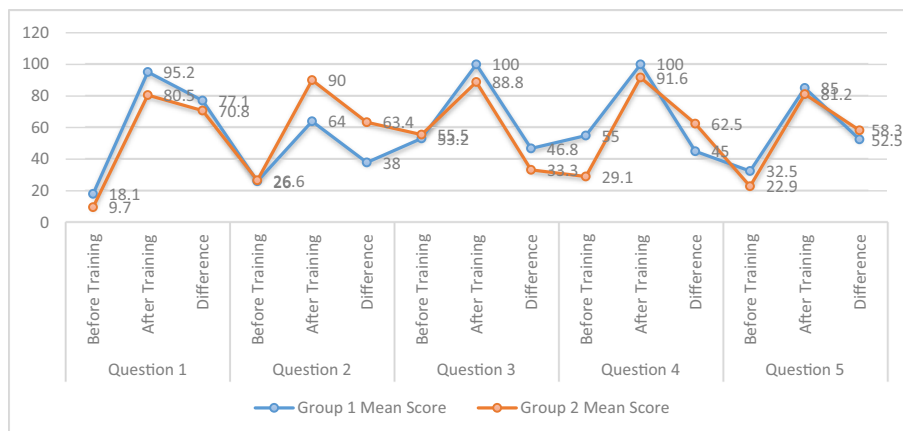
2.2.3 Perspective form

A protocol consisting of open-ended questions was developed to be used at the end of the training for the perspective dimension of computational thinking. While creating the questions, the sub-dimensions of the perspective dimension - benefit, competence and collaboration - were taken as the bases. As the questions were specific to prospective teachers, the sub-dimension of benefit was detailed in terms of professional life and thinking skills, and the sub-dimension of competence was in terms of reflection and estimation.

The data obtained from the protocol for the perspective dimension were analysed through descriptive and content analysis. The codes were analysed by two different experts and the data were presented transparently for reliability. Accordingly, the data were clarified in terms of benefit, competence and collaboration sub-dimensions of perspective dimension of computational thinking.

3 Results

The results of the study were given as follows with respect to skill test, graduation projects and perspective results.



Graph 1 Computational thinking skill test results of the participants before and after training

3.1 Results related to the first research question

The skill test, which includes the dimensions of computational thinking, was applied to prospective teachers before and after the training. The obtained data were analysed according to both analytical rubric and SOLO taxonomy.

According to Graph 1, it is observed that both groups had low pre-training scores for the first question which includes the algorithm and the highest increase after the training was therein. It can be explained by the fact that the prospective elementary teachers have not received training pertaining to programming or algorithm beforehand. In the second question, solution steps for a problem related to professional life were asked. In this question which includes all implementation steps of computational thinking, it was proved that the scores of both groups were the same before the training but the second group was more successful after the training. This difference, which appears in the second group, can be explained by the development of the program design with the design-based method. The fifth question includes fragmentation, abstraction, pattern, algorithm, evaluation and debugging dimensions of computational thinking, as the second question. The second group’s post-training score change was higher in this question like the second question.

There were similar increases in the pre- and post-training scores of the third and fourth questions in both groups, including the dimensions of pattern and abstraction. While the first group had higher scores in both questions after the training, the second group had higher increase in scores for the fourth question.

3.1.1 Computational thinking skill test by SOLO taxonomy

In SOLO taxonomy analysis of the computational thinking skill test, 10 out of 11 participants reached the extended abstract structure in the first question (Table 2). In the fifth question, where significant differences were observed before and after the training, it was also established that while pre-structural level is predominant before the training, they achieved at least “multi-structural level” after the training. In the second question, it was concluded that all the members of the second group reached the extended

Table 2 Computational thinking skill test results by SOLO taxonomy results

Before Training	After Training	Question 1		Question 2		Question 3		Question 4		Question 5	
		G1	G2	G1	G2	G1	G2	G1	G2	G1	G2
Pre-structural level	Relational level							1		2	2
Pre-structural level	Extended abstract level	1	3							2	1
Pre-structural level	Multi-structural level										2
Uni-structural level	Relational level		1			1	1	1	1		1
Uni-structural level	Extended abstract level	2			4	1	1		2		
Multi-structural level	Relational level			2		1	2	1	3		
Multi-structural level	Extended abstract level	2	2	3	2	1	2	2		1	
Relational level	Extended abstract level					1					

abstract structure level and about half of the first group achieved at least the relational structure after the training.

3.2 Results related to the second research question

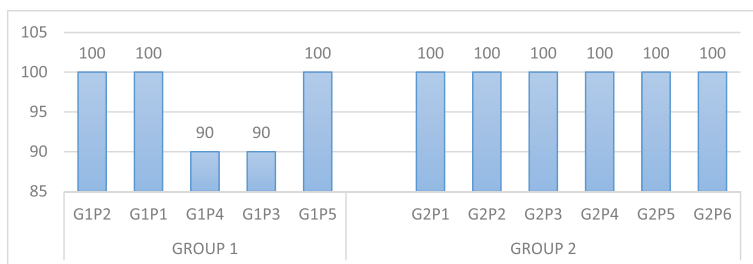
Prospective teachers were asked to complete their projects related to daily or professional lives according to the required criteria in the dimensions of knowledge and implementation. The projects in the knowledge dimension are intended to create direct algorithm and the respondents were informed that they can arbitrarily write pseudo codes in addition to flowchart. For the projects in the implementation step, they were asked to handle the relevant problem situation according to the steps of decomposition, abstraction, pattern / model extraction, algorithm, evaluating and debugging.

When the knowledge dimension projects including algorithm developed by the prospective teachers are examined, it is observed that all the projects are related to the situations that can be encountered in daily life or devices. In one of the projects, it was demonstrated that the algorithm on the bus cards that people use ordinarily on a daily basis was not as simple as the use of the ticket. In another one, they explained the working principle of recent robot cashiers in the markets with the algorithm. They also displayed the steps that take place on the screens and in the background of the devices through algorithms such as check-in via the Internet, making appointments from the hospital or shopping online.

The evaluation of participants' projects in the knowledge dimension included the criteria of understanding of the problem, operator use, use of loops, condition use, and flowchart use. It was ascertained that the prospective teachers in both groups prepared their projects according to the criteria and consequently got high scores (Graph 2). The fact that the two participants in the first group lack in using the flowchart may stem from the smaller number of algorithm activities than the second group.

When we look at the projects prepared for the implementation dimension of computational thinking, it is possible to see examples from daily routines to the technological devices encountered in daily life (Table 3). In these projects, the implementation dimension was requested to be explained in all its sub-dimensions such as abstraction, model extraction and pattern.

As it is a design-based study, some modifications have been made in the problematic areas of the program design of the second group and it has been observed that student achievement has increased (Graph 3). While the mean score of the first group of



Graph 2 Evaluation scores of computational thinking graduation projects in the dimension of knowledge

Table 3 Computational thinking graduation projects of prospective teachers in the dimension of implementation

Group 1		Group 2	
Participant	Project name	Participant	Project name
G1P1	Finding a Suitable Job for Waiters	G2P1	Friday Prayer
G1P2	Airline transporting	G2P2	Choosing a Hotel for the Holiday
G1P3	Borrowing Books from the Library	G2P3	Selecting Elective Courses at University
G1P4	Buying a Computer	G2P4	Selecting a Book
G1P5	Technological Polling	G2P5	Buying a Car
		G2P6	Cargo Vehicle System

implementation projects is 85 (according to absolute achievement percentage), the mean of the second group is 95.8. This supports the accuracy of the modification of the program.

3.3 Results related to the third research question

Prospective teachers' perspectives were evaluated under the headings of benefit, competence and collaboration. The heading of the benefit was discussed with regard to professional life and thinking skills (Table 4).

They stated that they can use computational thinking skills in every lesson, including branch courses, but they can also use it to solve the problems they encounter in their professional life and the student-related problems. In addition to the emphasis on the importance of understanding the problem, it was also noted that it contributes to become a qualified teacher. All of the prospective teachers declared that they benefited from the training in terms of problem solving skill among thinking skills. It was followed by the ability to question and noticing the details. Multilateral thinking and establishing a cause-effect relationship were also found useful as other thinking skills. They expressed that they can use computational thinking skills in daily life and social relations under the heading of reflection of the competency dimension. In the dimension of estimation, they voiced that they can estimate technological devices and the events in daily life. In the dimension of collaboration, prospective teachers want to take part in the projects and share the materials they produce in this field.

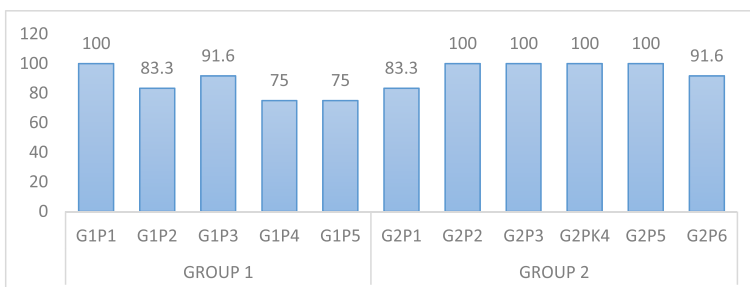
**Graph 3** Evaluation scores of computational thinking graduation projects in the dimension of implementation

Table 4 Perspectives of participants related to computational thinking

	Obtained Codes	Direct Quotations	
Benefit	Teacher Professional Life	G1P4 <i>"What I've learned has contributed me a lot. Most importantly, I realized the importance of understanding the problem to solve it. Acquiring this ability has changed the way I look at the problems that I face both in my daily life and my professional life."</i>	
	To be able to use in each lesson (5)		
	To be able to use in branch courses (4)		
	Importance of understanding the problem (4)		
	Solving the problems in professional life (2)		
	Thinking Skills	Becoming a qualified teacher (3)	G2P6 <i>"I think my problem-solving ability has improved. I started thinking mathematically because when I look at something, I start logical testing. I start from scratch. How did they think of it? How is its programming? It's not computer-based; to improve thinking, mental ability... I started to evaluate myself; it contributed me a lot in my daily life."</i>
		To be able to solve student-related problems (2)	
		Problem solving skill (11)	
		Learning questioning (9)	
		Multilateral thinking (5)	
Competence	Noticing the details (10)	G2P5 <i>"It's been everywhere in our daily lives, but we don't see it, it's everywhere. Recently, at a big supermarket, there's a robot in the quick cash point, so there's a perfect program in it. Now that you look at everything in that way, it's really precious."</i>	
	Change in perspective (5)		
	Establishing a cause and effect relationship (4)		
	To be able to design activities in one's own field (9)		
	To be able to use this skill in social relations (3)		
	Increased self-confidence (4)		
	Ability to use in daily life (10)		
	Estimating technological devices (10)		
	Estimating events / workflows in daily life (6)		
	Estimating intelligent systems (3)		
Collaboration	Taking part in project groups in this field (11)	G1P3: <i>"Actually, I've become thinking about programming of everything in my life. My perspective on these devices has changed, I started questioning. I realized that the products that have the capacity to produce are in exaggerated delivery to people, but they actually come out of a combination of several available features."</i> G1P3 <i>"...as this area is new and I believe the more I know, the more it can level me up, I would like to participate in all the activities that I can. I would also like to exhibit and share because I think other individuals should also be aware of the processes."</i>	
	Working with engineers (7)		
	Sharing one's products (3)		

4 Discussion

It was found that the program design was effective in both groups in this study in which the effects of computational thinking skills on prospective teachers were investigated. The parallelism of the results of the skill test and those of the graduation projects indicates that the research results are consistent with each other. It was detected that the means of the first, second and fifth questions in the skill test, which had low means before the training, increased after the intervention. The difference of the points the second group got before and after the training from the second and the fifth questions of the computational thinking skill test which involved the same dimensions of computational thinking but one was related to professional life and the other was with visual content, was higher than the first group. This situation shows the consistency of the dimensions included in the questions, and also shows that the updates made in the program design developed by the design-based method yielded positive results. Considering that block-based applications or computer-aided sources are generally used in the teaching and measurement of algorithms (Kalelioğlu et al. 2016), our study signals that it can also be used to teach algorithm and computational thinking skills with unplugged activities. The relatively small number of computer-aided and robotic activities supported different uses of computational thinking in these areas.

The results of skill test by analytical rubric are consistent with those by SOLO taxonomy. It can be asserted that SOLO taxonomy gives more concrete and more classified information for the level of acquisition of some skills. For instance, the score which regarded to be high can be equal to the relational or the extended abstract levels. It can be claimed that SOLO taxonomy, which is preferred in higher education (Ari 2013), is suitable for explaining the level of computational thinking.

Based on the point of view that computational thinking skills are necessary in every field from daily life to professional life (Wing 2006), program design for prospective teachers developed according to the IDC theory which proposes the use of content designed to address the problems of everyday life and the interests of learners (Chan et al. 2015). Kong (2016) stated that the use of this theory in curricula will provide an interesting theoretical basis. When the graduation projects were examined after the training, it was observed that the prospective teachers of both groups were successful in reflecting the competencies in accordance with the training they received. It is determined that the subjects chosen by the participants in their graduation projects range from the problems of daily life to technology and programming of intelligent systems. The variety of subjects in the prospective teachers' graduation projects coincides with the content in the program design which was prepared according to IDC theory. This situation is consistent with the scores of prospective teachers got from different types of questions (such as working principle of a device, professional life or visual questions) in the skill test. It also shows that they can use their computational thinking skills in various fields. According to Jonassen (2011), authentic problems inspired by everyday life are of greater interest to students, and so the students can engage more in problems and be effective in problem solving. This can be explained both by unplugged activities in the majority and by the development of content according to IDC theory.

When the changes in the perspectives of prospective teachers after the training were examined, it was concluded that the result was in positive manner. In addition, all the participants think that it is useful in developing problem solving skills, which is the

main purpose of computational thinking. It has been found that programming education improves problem solving skills (Dasso et al. 2005; Papert 1993), and there is a high correlation between computational thinking and problem solving (Román-González et al. 2017). Accordingly, it supports the view that computational thinking is a kind of problem solving skill (Wing 2006). The participants stated that there was also an improvement in their thinking skills in terms of detailed thinking and questioning and explained that they were able to reflect their acquired skills in daily life as well as in professional life as seen in the graduation projects. Questioning, which is expressed as one of the most important strategies used by teachers, is effective in learning and teaching processes (Tanisli 2013; Cotton 1989) and it is possible to understand the depth of the student's comprehension level with an effective inquiry (Moyer and Milewicz 2002). Due to the contribution of inquiry to teacher education, the relationship between computational thinking and inquiry can also be investigated. In addition to using computational thinking skills in their own courses, they stated that they could design activities related to this field. Prospective teachers who encountered algorithms and computational thinking skills for the first time expressed that they can estimate the working principles of technological devices. The participants made sense of the algorithms in the training they received and showed that they were able to transfer their skills to different areas in the graduation projects which are recommended in computational thinking (Brennan and Resnick 2012; Kong 2016).

5 Conclusion

In this study, the effects of a program design developed to provide prospective teachers with computational thinking skills on learners was examined. In the study where design-based method was used, program design was developed by focusing on subjects that will be of interest to the participants and the problems of daily life according to IDC theory. During the program development process, two separate groups of prospective elementary teachers were involved, who have not previously received programming or computational thinking training. The effects of program design, the vast majority of which was composed of unplugged activities, on prospective teachers were evaluated through pre-test-post-test computational thinking skill test and graduation projects.

As a result of the study, the following results were obtained.

- According to the analysis results of skill test applied to be pre-test and post-test by analytical rubric, it was determined that the computational thinking levels of learners increased in all the sub-steps of computational thinking knowledge and implementation dimensions.
- According to the analysis results of skill test applied for pre-test and post-test by SOLO taxonomy, it was confirmed that the learners in both groups reached to the relational and extended abstract structures namely top levels in computational thinking.

- According to the perspectives of students on computational thinking, it was observed that problem solving and questioning skills were exclusively developed and they could reflect the acquired knowledge and skills to daily and professional lives.
- The graduation projects prepared by the participants confirmed their perspectives. It can be claimed that the prospective teachers' ability to choose a wide range of subject areas from the problems of daily life to technological devices in their graduation projects is the outcome of the development of the program design according to IDC theory to some extent.

This study reveals that unplugged activities can be used to gain computational thinking skills and it can be realized through a program other than information technologies course. Therefore, it is significant to acquire computational thinking skills in a way that can be used in most areas of life. It is thought that it will be effective to develop the content according to IDC theory in this process.

References

- Arı, A. (2013). Bilişsel alan sınıflamasında yenilenmiş Bloom, SOLO, Fink, Dettmer taksonomileri ve uluslararası alanda tanınma durumları.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54.
- Bell, P. (2004). On the theoretical breadth of design-based research in education. *Educational Psychologist*, 39(4), 243–253.
- Booth W. A., (2013). *Mixed-methods study of the impact of a computational thinking course on student attitudes about technology and computation*. Unpublished doctoral dissertation, Baylor University.
- Brennan, K. & Resnick, M., (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver*, 1–25.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.
- Chan, T. W., Looi, C. K., & Chang, B. (2015). The IDC theory: Creation and the creation loop. Workshop proceedings of the 23rd international conference on computers in education (pp. 814–820).
- Cotton, K. (1989). Classroom questioning. Close-up No. 5. Portland, OR: Northwest Regional Educational Laboratory
- CSTA (2017). CSTA K-12 Computer Science Standards Revised. Retrieved December 20 2017 from <https://www.doe.k12.de.us/cms/lib/DE01922744/Centricity/Domain/176/CSTA%20Computer%20Science%20Standards%20Revised%202017.pdf>
- Curzon, P., McOwan, P. W., Plant, N., & Meagher, L. R. (2014). Introducing teachers to computational thinking using unplugged storytelling. In *Proceedings of the 9th workshop in primary and secondary computing education* (pp. 89–92).
- Dasso, A., Funes, A., Riesco, D., Montejano, G., Peralta, M. & Salgado, C. (2005). Teaching programming. *Proceedings of JEITICS, Educación en Informática y TICs en Argentina*, 183–186.
- Denning, P. J. (2009). The profession of it - beyond computational thinking. *Communications of the ACM*, 52(6), 28–29.
- Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, 32(1), 5–8.
- Google Education. (2017). Computational Thinking for Educators. Retrieved July 17, 2017 from <https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Hu, C. (2011). Computational thinking: What it might mean and what we might do about it. In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ACM, 223–227.

- 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.
- ISTE. (2016). *ISTE Standards For Students (Permitted Educational Use)*. Retrieved May 19, 2017, from www.iste.org/standards
- ISTE (2011). Computational Thinking Teacher Resources. Retrieved October 18, 2019 from https://www.iste.org/docs/ct-documents/ct-teacherresources_2ed-pdf.pdf?sfvrsn=2
- ISTE. (2018). *ISTE Standards For Educators: Computational Thinking Competencies*. Retrieved January 20, 2019, from www.iste.org/standards
- ISTE (2019). ISTE Computational Competencies. Retrieved 20 January, 2019 from <https://www.iste.org/standards/computational-thinking>
- ISTE & CSTA. (2011). *Computational Thinking Teacher Resources*. Retrieved October 18, 2018 from https://www.iste.org/docs/ct-documents/ct-teacher-resources_2ed-pdf.pdf?sfvrsn=2
- Jonassen, D. (2011). *Learning to solve problems: A handbook for designing problem-solving learning environments*. New York: Routledge.
- K-12 Computer Science Framework. (2016). Retrieved September 20, 2018 from <https://k12cs.org/>.
- Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, *4*(3), 583–596.
- Kong, S. C. (2016). A framework of curriculum design for computational thinking development in K-12 education. *Journal of Computers in Education*, *3*(4), 377–394.
- Korkut, F. (2002). Lise öğrencilerinin problem çözme becerileri. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, *22*(23), 177–184.
- Kotluk, N., & Kocakaya, S. (2015). 21.yüzyıl becerilerinin gelişiminde dijital öykülemeler: ortaöğretim öğrencilerinin görüşlerinin incelenmesi. *Journal of Research in Education and Teaching*, *4*(2), 354–363.
- Kuzu, A., Çankaya, S., & Mısırlı, Z. A. (2011). Tasarım tabanlı araştırma ve öğrenme ortamlarının tasarımı ve geliştirilmesinde kullanımı. *Anadolu Journal of Educational Sciences International*, *1*(1).
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, *2*(1), 32–37.
- LEGO Education. (2018a). WeDo 2.0 Computational thinking teachers guide.
- LEGO Education. (2018b). *About us*. Retrieved 17 December 2018 from <https://education.lego.com/en-us/about-us>
- Liu, Z., Zhi, R., Hicks, A., & Barnes, T. (2017). Understanding problem solving behavior of 6–8 graders in a debugging game. *Computer Science Education*, *27*(1), 1–29.
- Lu, J., & Fletcher, G. H. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, *41*(1), 260–264.
- 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.
- Moyer, P. S., & Milewicz, E. (2002). Learning to question: Categories of questioning used by preservice teachers during diagnostic mathematics interviews. *Journal of Mathematics Teacher Education*, *5*(4), 293–315.
- OECD, (2018). The future of education and skills Education 2030. [https://www.oecd.org/education/2030/E2030%20Position%20Paper%20\(05.04.2018\).pdf](https://www.oecd.org/education/2030/E2030%20Position%20Paper%20(05.04.2018).pdf)
- Özsoy, G. (2014). Problem çözme becerisi ile matematik başarısı arasındaki ilişki. *Gazi Üniversitesi Gazi Eğitim Fakültesi Dergisi*, *25*(3), 179–190.
- Papert, S. (1980). *Mindstorms: Children, computers and powerful ideas*. New York: Basic Books.
- Papert, S. (1993). *The Children's machine: Rethinking School in the age of the computer*. New York: Basic Books.
- Partnership for 21st Century Skills. (2015). *P21 Framework Definitions*. Retrieved October 05, 2018 from http://www.p21.org/storage/documents/docs/P21_Framework_Definitions_New_Logo_2015.pdf
- Qualls, J. A., & Sherrell, L. B. (2010). Why computational thinking should be integrated into the curriculum. *Journal of Computing Sciences in Colleges*, *25*(5), 66–71.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. B. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67.
- Richey, R. C., & Klein, J. D. (2014). *Design and development research: Methods, strategies, and issues*. New York: Routledge. <https://doi.org/10.4324/9780203826034>.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, *72*, 678–691.

- Selby, C.C. & Woollard, J., (2013). Computational thinking: The developing definition. *University of Southampton (E-prints) 1-6*. Retrieved December 20, 2018 from <https://eprints.soton.ac.uk/356481/>
- Tanisli, D. (2013). Preservice primary school mathematics teachers' questioning skills and knowledge of students in terms of pedagogical content knowledge. *Education and Science*, 38(169). Retrieved 20 December, 2019 from <http://egitimvebilim.ted.org.tr/index.php/EB/article/view/1736>
- Voogt, J. & Roblin, P. N. (2010). 21st century skills: discussion paper. Retrieved January 20 , 2019 from http://opite.pbworks.com/w/file/attach/61995295/White%20Paper%2021stCS_Final_ENG_def2.pdf
- Weiser, M. (1982). Programmers use slices when debugging. *Communications of the ACM*, 25(7), 446–452.
- Wiggins, G., & McTighe, J. (1998). What is backward design. *Understanding by Design*, 1, 7–19.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725.
- Wing, J. M. (2011). Research notebook: Computational thinking—What and why? *The link magazine*, spring 2011. Carnegie Mellon University, Pittsburgh. Retrieved February 27, 2018 from <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Ozlem Uzumcu¹ · Erdal Bay²

¹ Department of Primary Education Teaching, Faculty of Education, Hasan Kalyoncu University, Gaziantep, Turkey

² Department of Education Science, Gaziantep Islam Science and Technology University, Gaziantep, Turkey