Informatics in Education, 2019, Vol. 18, No. 1, 151–164

DOI: 10.15388/infedu.2019.07

Computational Thinking Self-Efficacy Scale: Development, Validity and Reliability

Volkan KUKUL[1], Serçin KARATAŞ[2]

[1]*Amasya University, Faculty of Education, Department of Computer Education and Instructional Technologies*

[2]*Gazi University, Faculty of Education, Department of Computer Education and Instructional Technologies*

e-mail: volkankukul@hotmail.com, sercin@gazi.edu.tr

Abstract. The aim of this study is to develop a self-efficacy measuring tool that can predict the computational thinking skill that is seen as one of the 21st century's skills. According to literature review, an item pool was established and expert opinion was consulted for the created item pool. The study group of this study consists of 319 students educated at the level of secondary school. As a result of the exploratory factor analysis, the scale consisted of 18 items under four factors. The factors are Reasoning, Abstraction, Decomposition and Generalization. As a result of applied reliability analysis, the Cronbach Alpha reliability coefficient can be seen to be calculated as .884 for the whole self-efficacy scale consisting of 18 items. Confirmative factor analysis results and fit indexes were checked, and fit indexes of the scale were seen to have good and acceptable fits. Based on these findings, the Computational Thinking Self-efficacy Scale is a valid and reliable tool that may be used in measuring to predict Computational Thinking.

Keywords: computational thinking, self-efficacy, programming and programming languages, assessment methodologies.

1. Introduction

Computers have become an irreplaceable tool in most people's daily lives. Many real-world problems are solved by means of computers or using the principles of computer science. For this reason, an individual's ability to use computers effectively is a skill that may help them to manage problems in their day-to-day activities (Booth, 2013; Barr, Harrison & Conery, 2011). However, people in technologically-developed countries may experience difficulties in solving problems using computers if they do not have a robust technological background. These kinds of people may not know how to make their lives easier by using technology effectively and in a way that enhances their quality of life (Wing, 2008; Computer Science Teacher Association [CSTA], 2011; National Research Council [NRC], 2010).

At present, people living in a technologically-enriched society need three different skills in order to be able to keep up with that society. The first of these is the ability to use basic computer applications. This skill is generally defined as *computer literacy*. The second is the ability to understand how computer systems work. This is generally defined as *computer fluency*. The third skill is the ability to use computer techniques or applications to solve specific problems. This skill is currently known as *computational thinking* (Perkovic and Settle, 2010). According to Wing (2006), computational thinking is a way of designing a system and understanding human behaviors by focusing on the basic concepts of the computer sciences.

Computational thinking as a problem-solving process promotes skills such as formulating problems, breaking down complex issues to find the main ideas and finding a solution using computers. While these skills are also used in problem-solving strategies other than computational problem-solving, a strong aspect of computational thinking is that it reduces the burdens on people through the use of technological devices (Booth, 2013). Nevertheless, computational thinking is not merely a skill used for solving problems involved with the programming of technological devices. Wing (2006) highlights the fact that computational thinking is not computer programming. In the same study, Wing also states that computational thinking does not mean "thinking like computers" and adds: "Computers are emotionless and boring; humans are smart and have strong imaginations. We humans make computers exciting."

In cases when what Wing highlights is not considered, it is inevitable that computational thinking is perceived solely as programming. This perception supports the idea that "If we teach programming to people, their computational thinking skills will improve", and it conflicts with Wing's definition. Thus, one of the most important points to be considered is that computational thinking differs from programming skills (Qualls and Sherrell, 2010). While programming is a process used by computer scientists or programmers to solve certain problems, computational thinking does not only involve programming, but also an acquired point of view which understands the benefits provided by technology for solving problems. Thus, although computational thinking may be a key requirement in programming, it is not limited to computer science (Denning, 2009).

When computational thinking is treated as a problem-solving process, it can be seen that it contains other processes within it, such as breaking down problems and solving each part one by one and/or modeling the problem coherently (Astrachan, 2009). In a study conducted by Hong Qin (2009) highlighting computational thinking as problem-solving, it was concluded that students used common aspects of their thinking processes to solve different problems and that they understood the importance of thinking beyond problem-solving. The '21st century student standards' determined by the International Society for Technology in Education (ISTE) include high-level skills such as problem-solving, cooperation, creativity and critical thinking. According to Brichacek (2014), computational thinking is directly related to these standards. If we apply computational thinking to high-level skills such as problem-solving, creativity and critical thinking, our capacity for innovation is likely to increase. Given that innovation is one of the skills required for students in the 21st century, it can be stated that computational thinking has a crucial role to play in delivering the standards expected in this century. In this regard, computational thinking should be among the key 21st century skills (Philips, 2009; Wing, 2010). Hence, Wing (2006) states that, like literacy, computational thinking should be a skill that everyone possesses. A person with computational thinking skills will demonstrate the ability to:

- Understand which aspects of a problem may be solved with digital tools such as computers.
- Evaluate the limitations and stronger aspects of digital tools and techniques for solving a problem.
- Apply a digital application or technique to a new problem.
- Understand whether there is an opportunity to use a new digital tool or technique to solve the problem.
- Apply digital strategies to different areas (cited from Cuny, Snyder & Wing by Wing, 2010).

With the understanding of education changing all over the world, it has become the primary purpose of education to produce innovative, creative, problem-solving and productive individuals. It is believed that students should have skills that allow them to keep up with modern society. Computational thinking is one of these skills and is seen to be as much a part of education as reading and writing by some researchers, such as Wing; they also argue that it is a crucial skill (Bundy, 2007; Day, 2011). Given the recent increase in the number of studies on computational thinking and in the grants provided by different organizations for research relating to computational thinking, it can be concluded that this concept is a significant one. It is important that educators enable students to acquire computational thinking skills from the primary/elementary school stage rather than waiting until high school or university, so that students are able to keep up with the modern world (Qualls and Sherrell, 2010).

While attempts have been made to help students to acquire this skill in educational environments, measurement procedures and measurement tools are required to assess whether this skill has been acquired or not. In this regard, it is important to measure the concept of self-efficacy (Zimmerman, 1995) in relation to concepts such as academic success, motivation, educational self-regulation. Although it is known that a number of factors have an effect on the success of learners, self-efficacy and attitude have been shown to be more important than other factors (Austin, 1987; Anastasiadou and Karakos, 2011). Self-efficacy may be defined as how individuals perceive their present ability to use a skill or achieve a goal (Bandura, 1977). A belief in self-efficacy is an individual's belief about whether s/he can exhibit the behaviors necessary in order to reach a desired target (Akbaş and Çelikkaleli, 2006). When examined within the scope of social cognitive theory, self-efficacy is considered to have an important role in revealing the emotions which have an effect on an individual's performance (Bandura, 2001). Self-efficacy is directly affected by the choices people make and their cognitive and affective processes and motivation (Bandura, 1995, p. 5). People who believe that they will be successful are individuals with high self-efficacy (Bandura, 1995, p. 10). Individuals who believe in their abilities outperform other individuals in difficult tasks. A high level of self-efficacy increases the number of successful outcomes and the success-related happiness of individuals (Bandura, 2001). In other words, it may be said that self-efficacy positively affects success, despite not being sufficient to predict success by itself (Akkoyunlu and Kurbanoğlu, 2003). The main aim when measuring self-efficacy is to measure performances in carrying out a task, rather than measuring the personal features of individuals (Zimmerman, 2000). The measurement of self-efficacy may be regarded as a component which gives an idea of how successful individuals are likely to be. Measures may then be taken to enhance their success and performance (Askar and Davenport, 2009). Hence, the measurement of self-efficacy has come to be seen as important and a number of self-efficacy scales have been developed for many concepts in the educational field. Even, some scales related to computational thinking like programming were developed (Ramalingam and Wiedenbeck, 1998; Kukul, Gökçearslan, and Günbatar, 2017). These scales contain the items related to the programming languages or block programming. Since computational thinking is not programming (although it is related), computational thinking self-efficacy scale is needed. However, no scale was found for measuring computational thinking self-efficacy. It was thus necessary to develop a new scale for measuring computational thinking self-efficacy.

2. Method

The scale development steps of DeVellis (2003) were followed while constructing the Computational Thinking Self-efficacy Scale. The first step is deciding what is to be measured. As computational thinking itself is a high-level skill, it is hard to observe. However, observing the skills that may predict this skill, and measuring the perception of a person relating to the presence of these skills, may provide easier and more correct results. Hence, it was decided to measure computational thinking self-efficacy. It was found that no such scale had been developed for measuring computational thinking self-efficacy. Therefore, a new scale was required.

2.1. *Creation of the Item Pool*

Item pool creation was the second step in developing the scale. The literature was reviewed in order to create an item pool, and definitions of computational thinking made in the literature and research on computational thinking were reviewed. The studies of Wing (2006, 2008), who enabled the popularization of the concept in the second half of the early 2000s were also examined. In addition to these studies, the definition of computational thinking given by the CSTA and ISTE and its sub-dimensions as well as the definition given by the organization Computing at School (CAT) were taken into consideration. At the end of the necessary research, an item pool of 47 items was created. The items were sent to 11 experts to get their opinions. 6 of the experts were academics working in programming/coding education in the field of Computer and Teaching Technologies Education. Among the other experts, one was a language expert, one was an assessment and evaluation expert and three were teachers working in the Ministry of National Education. All items of the scale and the factors predicted for the items, as well as a form outlining the purpose, scope and target group of the scale were sent to the experts via e-mail in order to gain their opinions. The experts were asked to assess each item as “appropriate”, “inappropriate” or “improvable”, and to add an explanation in a separate section for the “inappropriate” or “improvable” items. In addition, a section was incorporated below the form in which the expert could add items, and the experts were asked to add any suggestions for items in this area.

The items were re-examined in accordance with the opinions from the experts. The items deemed to be proper to exclude from the scale were removed, and the items suggested were incorporated. Consequently, a 51-item Self-efficacy Scale was obtained which was arranged as 5-point Likert type (1 = “Completely Disagree”, 2 = “Disagree”, 3 = “Neutral”, 4 = “Agree”, 5 = “Agree Fully”). The scale was applied as a pilot to secondary school students who would not participate into the real application in order to check age level conformity. Then minor revisions were made in the items for points not understood.

2.2. *Participants*

During the scale development stage, 409 students who were at secondary school in the 2016–2017 academic year were targeted. The students with data losses and who did not have any programming/coding education were excluded from the analyses. The analyses were conducted on 319 students. Demographic information relating to the study group is given in Table 1 and Table 2.

Table 1
Demographic Information of Self-Efficacy Scale Study Group

		n	%
Gender	Female	153	48
	Male	166	52
Total		**319**	**100**
Grade	5th Grade	118	37
	6th Grade	189	59.2
	7th Grade	12	3.8
Total		**319**	**100**

Table 2
Programming/Coding Education Previously Received by Students

Programming/Coding	n
Scratch	223
Code.org	115
Small Basic	15
Alice	7
Other	7

On examination of Table 1 and Table 2, it can be seen that 153 (48%) of the students participating in the study were female and 166 (52%) were male. Considering the grades of the students, it can be seen that 118 (52%) were in the 5th Grade, 189 (59.2%) in the 6th Grade, and 12 (3.8%) were in the 7th Grade. Examining the students' programming/coding education it can be seen that 223 students had had programming/coding education with Scratch and 115 students with Code.org education. The number of the students educated in Small Basic was 15, the number of the students educated in Alice was 7, and the number of the students educated with tools other than these tools was 7. The reason that the sum of these numbers exceeded the total number of the students was that students had been educated in more than one programming/coding tool.

3. Results

Exploratory factor analysis was performed in order to determine whether the scale was reliable. The results of the Kaiser–Meyer–Olkin (KMO) and Bartlett Sphericity tests were examined in order to determine whether the data obtained from the scale was appropriate for exploratory factor analysis. The KMO value was found to be .936, and this value may be interpreted as showing that the conformity of the data structure was suitable for performing the factor analysis in terms of sample size (Leech, Barrett & Morgan, 2005; Şencan, 2005; Tavşancıl, 2005). On examination of the Bartlett test results (X^2: 6992.560, SD: 1275, $p < .01$), the obtained chi-square value was seen to be significant at the .01 level. This result indicates that the data came from multivariable normal distribution and thus the factor analysis could be continued.

Principal components analysis was selected as the factorization method and maximum variability (varimax) was selected from the vertical rotation methods with the aim of revealing the factor pattern of the Computational Thinking Self-efficacy Scale.

3.1. *Results Relating to Validity*

On examination of the item statistics as a consequence of the conducted factor analysis, the items that seemed to have close factor load values in more than one factor (Kline,1993; Tabachnick and Fidell, 2013) or items with item load values detected to be under .30 were exclude. The factor analysis was then repeated, and an 18-item form was obtained.

At the end of the analysis, it was seen that there were four components with an eigenvalue over 1 for 18 items essential for the analysis. The contribution of these components to the total variance was found to be 54.717%. The said four components were seen to contribute significantly when the importance of their contribution to total variance was considered by examining both the explained total variance table and the scree plot graph. Accordingly, a repetition of the analysis was not required.

The results of the Kaiser–Meyer–Olkin (KMO) and Bartlett Sphericity tests were examined again in order to determine whether the data obtained with the 18-item form was appropriate for exploratory factor analysis. The KMO value was found to be .904, and this value may be interpreted as showing that the conformity of the data structure was perfect for performing the factor analysis in terms of sample size (Leech, Barrett & Morgan, 2005; Şencan, 2005; Tavşancıl, 2005). On examination of the Bartlett test results (X^2: 1778.892, SD: 153, p < .01), the chi-square value obtained was seen to be significant at the .01 level. This result indicates that the data came from multivariable normal distribution and that the factor analysis may thus be continued. At the end of the factor analysis conducted with 18 items, results relating to the variance explained by the components were given in Table 3.

On examination of Table 3, it can be seen that the contribution of the factors to the total variance after rotation was 15.19 for the first factor, 14.72 for the second factor, 12.47 for the third factor and 12.33 for the fourth factor.

In the exploratory factor analysis, which was conducted with the aim of revealing the factor pattern of the Computational Thinking Self-efficacy scale, the acceptance level for the factor load value was determined to be .40. Accordingly, it was observed that no item had a factor load value under .40. The factor pattern relating to the scale, factor load values of the items and common factor variances are given in Table 4.

Table 3
Explained Total Variance

	Initial Eigenvalues			Rotation Sums of Squared Loadings			Values after Rotation		
Factor	Total	Contribution to Variance (%)	Cumulative (%)	Total	Contribution to Variance (%)	Cumulative (%)	Total	Contribution to Variance (%)	Cumulative (%)
1	6.178	34.322	34.322	6.178	34.322	34.322	2.735	15.195	15.195
2	1.322	7.347	41.669	1.322	7.347	41.669	2.650	14.720	29.915
3	1.286	7.143	48.812	1.286	7.143	48.812	2.245	12.470	42.385
4	1.063	5.905	54.717	1.063	5.905	54.717	2.220	12.332	54.717
5	.877	4.870	59.586						
6	.809	4.493	64.079						
7	.765	4.251	68.330						
8	.710	3.946	72.276						
9	.674	3.746	76.022						
10	.617	3.430	79.452						
11	.595	3.303	82.755						
12	.533	2.960	85.715						
13	.529	2.936	88.651						
14	.474	2.634	91.286						
15	.447	2.481	93.767						
16	.411	2.283	96.050						
17	.372	2.069	98.118						
18	.339	1.882	100.000						

Table 4

Computational Thinking Self-Efficacy Scale Factor Pattern

Items	Factor				Common Factor Variance
	Factor 1	Factor 2	Factor 3	Factor 4	
m4	.781				.686
m24	.630				.532
m5	.592				.548
m20	.587				.495
m22	.569				.459
m45		.731			.607
m32		.652			.566
m31		.640			.544
m51		.551			.477
m50		.478			.427
m29			.764		.654
m7			.735		.636
m15			.615		.439
m34			.546		.472
m16				.750	.689
m21				.716	.667
m27				.629	.514
m30				.601	.439

As seen in Table 4, the factor load values in sub-scales level varied between .569 and .781 for factor 1, between .478 and .731 for factor 2, between .546 and .764 for factor 3 and between .601 and .750 for factor 4. On examination of the factor load values in terms of magnitude, it is possible to classify the load values of the items except m50 from "good" to "perfect". Item m50 may be classified as "medium" (Comrey and Lee, 1992). Kim-Yin (2004) highlighted that sample size should be considered when deciding to exclude an item from the scale. Accordingly, an item with a 0.40 factor load should have a sample size of at least 200 (cited by Şencan, 2005). That there were 319 participants in this study shows that an adequate sample size was reached for m50.

Haynes, Richard and Kubany (1995) stated that one of the most frequently used methods for excluding inappropriate items from a scale, enhancing content validity and determining the validity of its scope was expert opinion. At the end of the analyses con-

Table 5

Examples from Items in the Scale

Factor	Item
Reasoning	I can decide whether the data to be used for the solution of the problem is adequate or not
Abstraction	I can make comments on the data used for the solution of the problem
Decomposition	If there are sub-problems in the problem, I can manage the solution processes of these sub-problems
Generalization	I can make connections between the current problem and previously encountered problems

ducted, expert opinions were sought for naming the factors and to enable scope validity. The naming of the factors presented was accepted by the experts. Accordingly, the factors were labeled “Reasoning”, “Abstraction”, “Generalization” and “Decomposition”. In terms of scope, the 18-item form was considered to enable scope validity for the age level. Table 5 presents item examples from the scale.

3.2. *Results Relating to Reliability*

Prior to the reliability studies of the scale, the results of the validity studies were examined. After sorting out the overlapping items and items with item load under .30, reliability studies were conducted with the remaining 18 items. In order to determine the distinctiveness of the items in the four-factor scale, the t-test was conducted for determining whether the difference between test and item mean points of the lower and the upper 27% groups was in accordance with the total points. On examination of test and item means of the lower and upper 27% groups, it was seen that the difference between the lower and the upper 27% groups in the whole test was significant. Also, on calculation of item mean points, it was seen that the mean points of the lower and the upper 27% groups differed significantly. Cronbach’s alpha reliability coefficient was used to test the internal consistency of the scale, and the reliability of the whole scale and the sub-factors were calculated. Reliability coefficients calculated for each factor and the whole scale are given in Table 6.

On examination of Table 6, as a result of applied reliability analysis, Cronbach Alpha reliability coefficient can be seen to be calculated as 0.884 for the whole self-efficacy scale consisting of 18 items; 0.772 for the Reasoning sub-scale consisting of 5 items; 0.774 for the Abstraction sub-scale consisting of 5 items; 0.701 for the Decomposition sub-scale consisting of 4 items; and 0.718 for the Generalization sub-scale consisting of 4 items.

3.3. *Results Relating to Confirmatory Factor Analysis*

The fit of model relating to the factor structure revealed as a result of Exploratory Factor Analysis (EFA) was tested with Confirmatory Factor Analysis (CFA). Conformity of the obtained model was tested with $\chi 2$/sd, RMSEA (Root Mean Square Error Ap-

Table 6
Reliability Coefficients Relating to Computational Thinking Self-Efficacy Scale

Scales	Number of Items	Cronbach’s Alpha Coefficient	Reliability Level
Computational Thinking Self-Efficacy Scale	18	.884	High
Reasoning	5	.772	High Enough
Abstraction	5	.774	High Enough
Decomposition	4	.701	High Enough
Generalization	4	.718	High Enough

proximation), NFI (Normed Fit Index), NNFI (Non-Normed Fit Index), RMR (Root Mean Square Residual), CFI (Comparative Fit Index), IFI (Incremental Fit Index), GFI (Goodness of Fit Index) and AGFI (Adjusted Goodness of Fit Index) fit criteria. Required modifications were made in accordance with the analysis results, it was detected for the fit of model that $\chi2/sd = 1.535$; RMSEA = 0.041; NFI = 0.892; RMR = 0.049; CFI = 0.959; IFI = 0.959; GFI = 0.959 and AGFI = 0.916.

For a goodness of fit index, 0 indicates that there is no fit between the data and the model, and 1 indicates that there is a full fit. When the value of an index is greater than 0.90 and close to 1, it may be said that fit in the data is almost enabled (Çerezci, 2010). Şimşek (2007) states that an $\chi2/sd$ value of or under 5 and an RMSEA value of or under 0.08 indicate good fit. Byrne (1998) states that RMR and SRMR values of or under 0.10 are required for good fit. Again, an IFI, CFI, NFI and NNFI greater than 0.90 indicates a good model. In addition, an AGFI of or greater than 0.80 and a GFI of or greater than 0.85 indicate acceptable fit (Çokluk *et al.*, 2010). Meydan and Şeşen (2011) utilized fit values indicated by different research in their study and tabulated statistical values relating to model fits (p. 37). Accepted intervals in accordance with this table and model fit values in accordance with the data obtained are given in Table 7.

On examination of Table 7, it can be seen that the Chi-Square Fit test result showed good fit. Fit indexes indicate that indexes other than NFI and CFI exhibited good fit. CFI is seen to be within the acceptable fit interval. As the NFI value was so close to the acceptable fit border, no modification was required.

There are different opinions relating to which fit indices are to be reported in the confirmatory factor analysis. While there is an agreement on reporting the chi-square fit test (Meydan and Şeşen, 2011; İlhan and Çetin, 2014), it is also deemed beneficial to state the results of a few other indexes (Meydan and Şeşen, 2011, p. 38). In this respect, the fit values reported in this study may be considered adequate. On examination of the goodness of fit indexes obtained in this study, the model fit of the scale is seen to be at a statistically acceptable level. The model formed as a result of amendments relating to the scale developed and the relations between the factors are given in Fig. 1.

Table 7

Statistical Values Relating to Structural Equality Model Fit and Fit Points of the Scale

Measurement (Fit Statistics)	Good Fit	Acceptable Fit	Computational Thinking Self-Efficacy Scale Fit Points	Fit Status
X^2	Insignificant	-	Insignificant	
X^2/sd	≤3	≤4–5	1.535	Good Fit
NFI	≥0.95	0.94–0.90	0.892	-
IFI	≥0.95	0.94–0.90	0.959	Good Fit
CFI	≥0.97	≥0.95	0.959	Acceptable Fit
RMSEA	≤0.05	0.06–0.08	0.041	Good Fit
GFI	≥0.90	0.89–0.85	0.959	Good Fit
AGFI	≥0.90	0.89–0.85	0.916	Good Fit
RMR	≤0.05	0.06–0.08	0.049	Good Fit

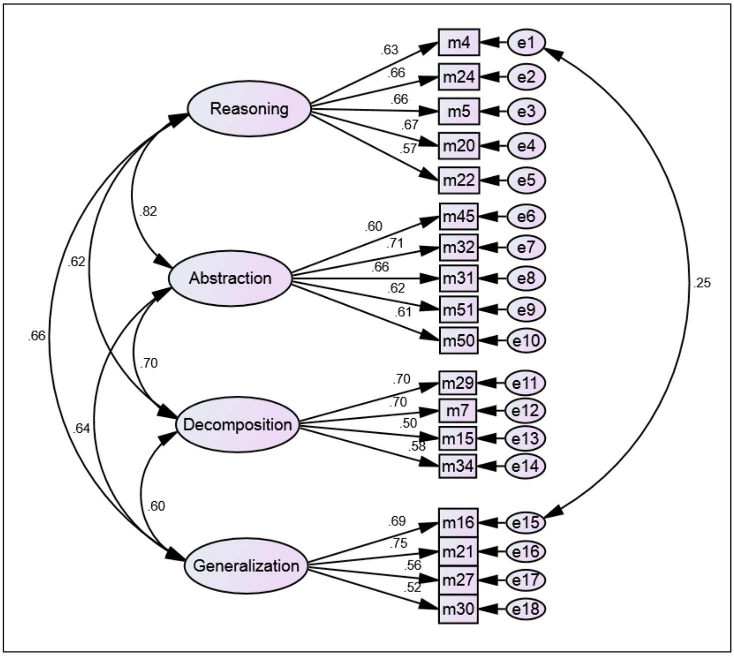

Fig. 1. Post-modification Structural Equality Model of the Scale.

On examination of Fig. 1, it can be seen that the standardized relationship values between factors were .60 and over. On examination of the relationship between items and factors, it was determined that values ranged from .50 to .75.

4. Conclusion

At the end of this study, a scale was developed which consists of 4 factors and 18 items for measuring the computational thinking self-efficacy of secondary school students. While developing the scale, the items were formed by utilizing the related literature and expert opinions. The scale was finalized with the application of exploratory and confirmative factor analyses. According to the results of the exploratory factor analysis, the scale explained 54.717% of the total variance. According to the reliability study conducted, the reliability coefficient of the whole scale was detected to be .884.

The reason behind the decrease in the number of items in the final form of the scale compared with the initial ones might be the close relationship between the CT concepts. Wing (2008) emphasizes that one of the two skills that is important for CT is Abstraction. For example, creating algorithms is a kind of abstraction. This result explains the absence of an algorithmic thinking factor on the scale. Because of the factor loadings of the items categorized under algorithmic thinking were higher in the abstraction factor, no separate algorithmic thinking factor was formed. A similar result was also observed for evaluation factor. According to the Computing at School (2014), although Evaluation is a sub-dimension of Computational Thinking, a separate factor was not constructed as the evaluation was carried out at every step of Computational Thinking. When the items classified in the first factor are analyzed, it was observed that these items are related with algorithmic thinking, evaluation, and the data that can be used for problem-solving. Cognitive factors are also known as effective in explaining CT skills. One of these factors is reasoning ability (Roman-Gonzalez, Perez-Gonzalez, Moreno-Leon, & Robles, 2018). When the items collected in the first factor were evaluated, it was seen that they were suitable for reasoning ability. At the end of the literature review, the sub-dimensions of the scale were given the headings "Reasoning", "Abstraction", "Decomposition" and "Generalization". It was decided that this nomenclature was appropriate in accordance with opinions from the experts.

Confirmatory factor analysis results and fit indexes were checked, and fit indexes of the scale were seen to have good fits, other than NFI ($\chi 2/sd = 1.535$; RMSEA = 0.041; NFI = 0.892; RMR = 0.049; CFI = 0.959; IFI = 0.959; GFI = 0.959; AGFI = 0.916). The fact that the NFI value was close to .90, the acceptable fit value, was interpreted as meaning that it could be ignored and modification was not required. These results can thus be interpreted to show that the scale was theoretically appropriate.

The scale that has been developed is considered to be of benefit for the measurement of computational thinking, a topic which has been intensively studied by educators recently. Considering the relationship between CT self-efficacy and CT skills (Roman-Gonzalez *et al*., 2018), this scale can be considered as a measurement tool that can help teachers develop their students' CT skills. Its structure validity could be tested by applying the scale to more secondary school students. In this way, a more valid scale could be obtained. In future research, a similar measurement tool with different items and for different age levels could be assessed. Studies could also be conducted on the relationship between different measurement tools developed to measure computational thinking and this self-efficacy scale. Results of this research may enable modification of the measurement tools developed and more accurate measurement of computational thinking.

5. References

Adams, J.B. (2008). Computational science as a twenty-first century discipline in the liberal arts. *Journal of Computing Sciences in Colleges,* 23(5), 15–23.

Akbaş, A., Çelikkaleli, Ö. (2006). Sınıf öğretmeni adaylarının fen öğretimi özyeterlik inançlarının cinsiyet, öğrenim türü ve üniversitelerine göre incelenmesi. *Mersin Üniversitesi Eğitim Fakültesi Dergisi*, 2(1).

Akkoyunlu, B., Kurbanoğlu, S. (2003). Öğretmen adaylarının bilgi okuryazarlığı ve bilgisayar öz-yeterlik algıları üzerine bir çalışma. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, 24(24).

Anastasiadou, S.D., Karakos, A.S. (2011). The beliefs of electrical and computer engineering students regarding computer programming. *The International Journal of Technology, Knowledge and Society*, 7(1), 37–51.

Astrachan, O. (2009). A New Way of Thinking About Computational Thinking. *ACM SIGCSE.* Chattanooga, ABD.

Askar, P., Davenport, D. (2009). An investigation of factors related to self-efficacy for java Programming among engineering students. *The Turkish Online Journal of Educational Technology TOJET*, 8(1): 26–32.

Austin, H.S. (1987). Predictors of pascal programming achievement for community college students. In: *Proceedings of the Eighteenth SIGCSE Technical Symposium on Computer Science Education*. Missouri, United States, 161–164. DOI: 10.1145/31726.31752.

Bandura, A. (1977). *Self-efficacy: Toward a unifying theory of behavioral change. Psychological Review*, 84, 191–215, http://dx.doi.org/10.1037/0033-295X.84.2.191

Bandura, A. (Ed.). (1995). *Self-efficacy in Changing Societies*. Cambridge university press.

Bandura, A. (2001). Social cognitive theory: An agentic perspective. *Annual Review of Psychology*, 52(1), 1–26. DOI: 10.1146/annurev.psych.52.1.1.

Barr, D., Harrison, J., Conery, L. (2011). *Computational Thinking: A Digital Age. Learning&Leading With Technology*, 20–23.

Booth, A.W. (2013). *Mixed-Methods Study of the Impact of a Computational Thinking Course on Student Attitudes about Technology and Computation*. Doktora Tezi, Baylor University, Department of Educational Psychology, Baylor.

Brichacek, A. (2014). Computational thinking boosts students' higher-order skills. https://www.iste.org/explore/articleDetail?articleid=232&category=Featured-videos&article=Computational%20thinking%20boosts%20students%E2%80%99%20higher-order%20skills adresinden 21.05.2015 tarihinde ulaşılmıştır

Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.

Büyüköztürk, Ş., Çakmak, E.K., Akgün, Ö.E., Karadeniz, Ş., Demirel, F. (2012). *Bilimsel Araştırma Yöntemleri*. Ankara: Pegem Akademi.

Byrne, B.M. (1998). *Structural Equation Modeling with Lisrel, Prelis and Simplis: Basic Concepts, Applications, and Programmings*. London: Lawrence Erlbaum Assocatiates, Publishers.

Comrey, A.L., Lee, H.B. (1992). *A First Course in Factor Analysis (2nd ed.)*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Creswell, J.W. (2012). *Educational Research Planning, Conducting, and Evaluating Quantitative and Qualitative Research.* Boston: Pearson.

Creswell, J.W., Plano Clark, V.L. (2015). *Karma Yöntem Araştırmaları Tasarımı ve Yrütülmesi.* (Y. Dede, S.B. Demir, Çev.) Ankara: Anı Yayıncılık.

CSTA, ISTE. (2011). *Computational Thinking in K–12 Education Leadership Toolkit.* Computer Science Teacher Association: http://csta.acm.org/Curriculum/sub/CurrFiles/471.11CTLeadershiptToolkit-SP-vF.pdf adresinden alındı

Çerezci, E.T. (2010). *Yapısal Eşitlik Modelleri ve Kullanılan Uyum Iyiliği Indekslerinin Karşılaştırılması*. (Unpublished Doctoral Dissertation). Gazi Üniversitesi Fen Bilimleri Enstitüsü: Ankara.

Day, C. (2011). Computational thinking is becoming one of the three rs. *Computing in Science & Engineering*, 13(1), 88–88.

Denning, P.J. (2009). The profession of it beyond computational thinking. *Communications of The ACM,* 52(6), 28–30.

DeVellis, R.F. (2003). *Scale Development: Theory and Applications (2nd ed.)*. Thousand Oaks, CA: Sage.

Haynes, S.N., Richard, D., Kubany, E.S. (1995). Content validity in psychological assessment: A functional approach to concepts and methods. *Psychological Assessment*, 7(3), 238–247.

Hemmendinger, D. (2010). A Plea for Modesty. *ACM Inroad,* 1(2), 4–7.

Kline, P. (1993). *An Easy Guide to Factor Analysis*. New York, NY: Routledge.

Kukul, V., Gökçearslan, Ş., Günbatar, M. S. (2017). Computer programming self-efficacy scale (CPSES) for secondary school students: Development, validation and reliability. *Eğitim Teknolojisi Kuram ve Uygulama*, 7(1).

İlhan, M., Çetin, B. (2014). LISREL ve AMOS programları kullanılarak gerçekleştirilen yapısal eşitlik modeli (yem) analizlerine ilişkin sonuçların karşılaştırılması. Eğitimde ve Psikolojide Ölçme ve Değerlendirme Dergisi, 5(2).

Leech, N.L., Barrett, K.C., Morgan, G.A. (2005). *SPSS for Intermediate Statistics: Use and Interpretation*. Psychology Press.

Meydan, C.H., Şeşen, H. (2011). *Yapısal Eşitlik Modellemesi AMOS Uygulamaları.* Ankara: Detay Yayincilik.

NRC. (2010). *Report of a Workshop on The Scope and Nature of Computational Thinking.* Washington D.C.: The National Academies Press.

Perkovic, L., Settle, A. (2010). Computational thinking across the curriculum: A conceptual framework. *College of Computing and Digital Media Technical Report,* 10-001.

Phillips, P. (2009). *Computational Thinking A Problem Solving Tool For Every Classroom.* Computer Science Teacher Association: http://csta.acm.org/Resources/sub/ResourceFiles/CompThinking.pdf adresinden alındı

Qin, H. (2009). Teaching computational thinking through bioinformatics to biology students. In: *40th ACM Technical Symposium on Computer Science Education*. New York,ABD, 188–191.

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.

Ramalingam, V., Wiedenbeck, S. (1998). Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research*, 19(4), 367–381.

Roman-Gonzalez, M., Perez-Gonzalez, J.-C., Moreno-Leon, J., Robles, G. (2018). Extending the nomological network of computational thinking with non-cognitive factors. *Computers in Human Behavior,* 80, 441–459.

Şencan, H. (2005). *Sosyal ve Davranışsal Ölçümlerde Güvenilirlik ve Geçerlilik*. Seçkin Yayıncılık.

Şimşek, Ö. F. (2007). *Yapısal Eşitlik Modellemesine Giriş, Temel Ilkeler ve LISREL Uygulamaları*. Ankara: Ekinoks Yayıncılık.

Tabachnick, B.G., Fidell, L.S. (2013). *Using Multivariate Statistics (6th ed.)*. Boston: Pearson.

Tavşancıl, E. (2005). *Tutumların Ölçülmesi SPSS Veri Analizi*. *Ankara: Nobel Yayınları*.

Yıldırım, A., Şimşek, H. (2011). *Sosyal Bilimlerde Nitel Araştırma Yöntemleri* (8. b.). Ankara: Seçkin.

Weinberg, A.E. (2013). *Computational Thinking: An Investigation Of The Existing Scholarship And Research.* Doktora Tezi, Colorado State University, School of Education, Colorado.

Wing, J. (2006). Computational Thinking. *Communications of The ACM,* 49(3), 33–35.

Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of The Royal Society*, 3717–3725.

Wing, J. (2010). *Computational Thinking: What and Why?* Center for Computational Thinking Carnegie Mellon: `https://www.cs.cmu.edu/~CompThink/papers/TheLinkWing.pdf adresinden alındı`

Zimmerman, B.J. (1995). Self-efficacy and educational development. *Self-efficacy in Changing Societies*, 202–231.

Zimmerman, B. J. (2000). Self-efficacy: An essential motive to learn. *Contemporary Educational Psychology*, 25(1), 82–91. DOI: 10.1006/ceps.1999.1016.

V. Kukul is a researcher in the Department of Computer Education and Instructional Technology at Amasya University. And also, he works in Distance Education Research and Application Center at the same university. He had his Ph.D. on teaching programming to kids. He is interested in distance education, computing education, instructional design and digital storytelling.

S. Karataş is a professor in the Department of Computer Education and Instructional Technology at Gazi University. Her interest areas are distance education theories and implementations, management of distance education, instructional technology, interaction in online learning environments, developing materials for online learning environments, creative thinking in learning and gamification in education.