01. Self-Assessment of Computational Thinking Skill: A Scale Development Study[1]

Veysel Karani CEYLAN[2]

Ruken AKAR VURAL[3]

APA: Ceylan, V. K. & Akar Vural, R. (2023). Self-assessment of computational thinking skill: A scale development study. *Rumeli Eğitim Araştırmaları Dergisi*, (4), 1-22. DOI: 10.5281/zenodo.8217317

Abstract

This study aims to develop a valid and reliable scale to measure computational thinking skills in children aged between 11 and 13. The scale development process began with an in-depth literature review to determine the operational definition of computational thinking. In addition, 17 teachers working in the field were interviewed to determine how the concept is used in practice. Based on the literature and interview data, the dimensions of algorithmic thinking, abstraction, reusability, automation, generalization, parsing, and parallelization were identified as sub-dimensions of the draft scale. The item pool was initially set at 40 items, and the opinions of subject experts were sought for content and face validity. After this stage, the item pool was reduced to 36 items. The pilot study of the draft scale form was administered to a total of 272 students. After the pilot application of the draft scale form, exploratory factor analysis was first used in the analysis phase. When the factor relations of the scale were determined as a result of repeated analysis, 16 items were removed from the scale, leaving 20 items in the final form. Cronbach alpha for internal consistency and reliability values indicated sufficient reliability values. Confirmatory factor analysis was then used to check the validity of the factors obtained. The compatibility of the 5 factors obtained as a result of EFA with the items was tested in CFA. As a result of the analysis, five factors were obtained: algorithmic thinking (5 items), parallelization (5 items), decomposition (4 items), automation (3 items), and abstraction (3 items) scale structure consisting of 20 items.

Keywords : Scale development , computational thinking, factor analyse

Bilgi işlemsel düşünme öz değerlendirme ölçeği: Ölçek geliştirme çalışması

Öz

Bu çalışmanın amacı, 11-13 yaş arası çocukların bilgi işlemsel düşünme becerilerini belirleyen geçerli ve güvenilir bir ölçek geliştirmektir. Ölçek geliştirme sürecinde öncelikle bilgi işlemsel düşünmenin tanımını belirlemek için derinlemesine bir literatür taraması yapılmıştır. Ayrıca alanda görev yapan 17 öğretmenle görüşülerek kavramın uygulamada nasıl kullanıldığı belirlenmiştir. Literatür ve görüşme verilerine dayanarak; taslak ölçeğin alt boyutları olarak algoritmik düşünme, soyutlama, yeniden kullanılabilirlik, otomasyon, genelleme, ayrıştırma ve paralelleştirme boyutları belirlenmiştir. Madde havuzu başlangıçta 40 madde olarak belirlenmiş, kapsam ve görünüş

[1] This study was derived from the doctoral dissertation titled "The Effect of Scenario-Based Scratch Curriculum on Students' Computational Thinking Skills, Problem Solving and Programming Unit Achievements" of first author.

[2] Dr. MEB (Aydın, Türkiye), vceylan@gmail.com, ORCID ID: 0000-0002-8899-960X [Araştırma makalesi, Makale kayıt tarihi: 15.04.2023-kabul tarihi: 02.06.2023; DOI: 10.5281/zenodo.8217317]

[3] Prof. Dr., Aydın Adnan Menderes Üniversitesi, Eğitim Fakültesi, Eğitim Bilimleri Bölümü (Aydın, Türkiye) rakarvural@gmail.com, ORCID ID: 0000-0002-3137-3753

geçerliliği için alan uzmanlarının görüşlerinden sonra 4 madde çıkarılmıştır. Taslak ölçek formunun pilot çalışması toplam 272 öğrenciye uygulanmıştır. Taslak ölçek formunun pilot uygulamasının ardından analiz aşamasında ilk olarak Açımlayıcı Faktör analizi kullanılmıştır. Tekrarlanan analizler sonucunda ölçeğin faktör ilişkileri net olarak belirlendiğinde 16 madde ölçekten çıkarılmış ve geriye 20 madde son halini almıştır. Cronbach alpha iç tutarlılık ve güvenilirlik değerleri, yeterli güvenilirlik değerlerine işaret etmektedir. Faktörlerin geçerliliği için doğrulayıcı faktör analizi kullanılmıştır. AFA sonucunda elde edilen 5 faktörün maddelerle uyumu DFA'da test edilmiştir. Analiz sonucunda algoritmik düşünme (5 madde), paralelleştirme (5 madde), ayrıştırma (4 madde), otomasyon (3 madde), soyutlama (3 madde) olmak üzere 20 maddeden oluşan beş faktörlü bir ölçek yapısı edilmiştir.

Anahtar kelimeler : Bilgi işlemsel düşünme, faktör analizi, ölçek geliştirme

1. Introduction

With the developments in the world of information and communication, information has increased significantly (big data) since the existence of humanity, and processing information with high-level thinking skills is considered a more important value. Computational Thinking (CT), which is shown among these thinking skills, is the most mentioned thinking area nowadays with its core skills (Brennan & Resnick, 2012). It is estimated that increasing these competencies of individuals will also be an important tool in creating value-added products since the development of ICT skills also enables the development of the transfer of knowledge in many areas (Garcīa-Penalvo et al., 2016). Several studies have been carried out on the interaction of CT skills with other thinking areas and it has been determined that they are positively related to areas such as problem-solving, reasoning, reflective thinking, and spatial thinking (Orton et al., 2016; Ambrosio et al., 2014; Selby & Woollard, 2013; Yıldız Durak & Sarıtepeci, 2018).

The basis of term CT is based on Papert's (1980) constructionism and APOS (Action Process Object Schema) theory. The term computer, whose English equivalent is "computer", was first used in English in the sense of a person who calculates in the 1640s, and the origin of this word comes from "Compute", that is, to calculate. The calculation, which originated in Turkish, has passed as information counting and has caused the problem of how to express it in Turkish. There is still no clear equivalent of CT in Turkish literature. In addition to its definition, it is seen that there are different approaches and uses related to this skill, although there is still no clarity in the literature in terms of core skills, pedagogical, measurement, and evaluation of the concept. The concept of CT is also defined in different ways in different studies. The most important common points of these definitions are; problem-solving processes such as problem-solving, understanding the problem, and formulating problems. In a joint study reported by CSTA (Computer Science Teachers Association), computational thinking was expressed as a problem-solving process with the following features:

- Formulating to solve problems with the help of computers or other tools,
- Organizing and analyzing data logically,
- Presenting data through models, simulations,
- Automating solutions within the framework of algorithmic thinking,
- Identifying, analyzing, and applying the most appropriate solution(s) by using resources effectively and efficiently.

- Transferring and generalizing the found solution to different problems (Gülbahar, Kert &, Kalelioğlu et al., 2014; ISTE , 2016).

1.1. Sub-Dimensions of Computational Thinking

The computational thinking process includes many sub-actions and concepts. In the literature review, it is seen that CT is not a single generally accepted component of core skills (Kalelioğlu et al., 2014). However, when the different views accepted in the literature are examined (ISTE, 2016; Kalelioğlu & Gülbahar, 2015; Selby & Woollard, 2013; Wing, 2006); abstraction, algorithmic thinking, evaluation, problem-solving, separation into components, pattern recognition and generalization. Selby and Woollard (2013) used the concepts of thinking process (Cuny et al., 2010), abstraction (Denning, 2007), and decomposition (Edelson, 2002; Guzdial et al., 2019) in their study on the definition of CT. Today, Microsoft and Google companies dealing with CT also examine these components under two headings as mental processes and concrete outputs. For mental processes; They identified the components of abstraction, algorithmic design, data analysis, decomposition, and pattern recognition. Concrete outputs were determined as automation, data collection, data representation, parallelization, generalization of patterns, and simulation. Sub-dimensions of CT frequently used in the literature are summarized in and explained in Table 1.

Table 1. Sub-dimensions of CT

	Concept
Abstraction	Abstraction is the process of making work more understandable by reducing unnecessary details. The important thing in the abstraction skill is to choose the right detail that needs to be hidden without losing the data so that the problem can be solved more easily. An important part of this is choosing a good representation of the system. Different presentations make it easier to do different things. (Csizmadia et al., 2015).
Algorithmic Thinking	Algorithmic thinking is the way to solve a clear definition of the processing steps (Csizmadia et al., 2015).
Decomposition	Decomposition is a way of thinking about artifacts in terms of their parts. The parts are then understood, deciphered, developed, and evaluated. This makes it easier to solve complex problems, new situations are better understood, and large systems are easier to design (Csizmadia et al., 2015).
Debugging	Debugging is the systematic practice of analysis and evaluation using skills such as testing, monitoring, and logical thinking to predict and verify results.
Generalization	Generalization is related to identifying patterns, similarities, and connections and using these features. It is a way to quickly solve new problems based on previous solutions to problems and new experiences. Algorithms that solve some specific problems can be adapted to solve an entire class of similar problems.

1.2. Measurement and evaluation status of CT

It is seen that there is no dominant model or approach in the literature review for the measurement and evaluation of CT skills. It is seen there are differences in the evaluation frameworks and models seen in the literature. In most of the existing approaches, it is seen that the tools used in the evaluation of the final products developed by the students with this thinking skill are insufficient.

Seiter and Foreman (2013) in the early-stage CT (PECT) model 1-6. They assume that every student naturally possesses CT skills in their approach to determining the CT levels of grade-level students and that these skills emerge when designing and developing specific tasks. There are three different levels of variables in the model; are variables that represent concepts, design patterns, and programming structure. Concept variables of the model for basic programming concepts; It creates concepts such as operators, conditions, views, loops, and variables. Design patterns, on the other hand, consist of concepts for the design of the developed product such as animation, speech, scenario, and score tracking. The variables that show the programming structure of the model consist of procedure, algorithm, parallelism, abstraction, and synchronization steps. Seiter and Foreman (2013) graded these variables as "Basic (1)", "Improvable (2)" and "Adequate (3)" and evaluated them over these units.

Grover, Cooper and Pea (2013) used open-ended questions, multiple-choice tests, and the applications they made to develop a certain algorithm on the Scratch program as an evaluation tool in their study on the evaluation of the concept of CT at the basic education level.

As another evaluation strategy, scenarios such as leaving certain points of the process blank or correcting the error by finding a faulty program piece by the student are used during the development of the product. In this way, the scenarios of finding the problems and the method of debugging the current program are shown among the effective methods in measuring the CT skills of the students (Repenning et al., 2010).

In addition to these models, the design scenarios suggested by Brennan and Resnick (2012) are; They developed different project sets by using the Scratch programming tool in collaboration with the Education Development Center (EDC). This model includes the steps of choosing one of the previously developed sets by the students, explaining the development steps of the selected project, how this project can be developed, debugging if any, and rearranging it by adding a new feature.

At the beginning of the studies on the evaluation of CT in an uncomputerized environment is the bebras.org event, whose popularity is increasing every year and whose participation level is increasing in our country. It is an activity aimed at increasing both CT skills and different thinking skills of individuals with multiple-choice answers consisting of short tasks that appeal to different age groups (Kalelioğlu et al., 2014).

Although many studies have been carried out on the evaluation of CT, the difficulty of concepts and structures in computer programming environments is an important obstacle in this process. In the context of measuring and evaluating CT skills, although there is not a single common scale widely used in the literature, many different tools can be accessed in this field. The most important obstacle encountered in the measurement of CT skills stems from the nature of computer science (Yeni, 2017). Because in this area, a question can sometimes have too many correct options. For this reason, it is seen that these scales are associated with the relevant tools in studies and different measurement tools are used instead of a single scale. At the beginning of these tools are rubrics to evaluate the sub-components of CT, online tools for coding analysis (Dr. Scratch), qualitative interviews with the participants, in-class observations, and a few scale studies developed in the field (Armoni et al., 2015; Field, 2009; Repenning et al., 2010; Werner et al., 2012). In addition to this, it is seen that there are studies on the transfer of these skills to different disciplines at the basic education level. Schwartz and Martin (2004) conducted studies in the field of mathematics and science at the secondary school level in their work titled preparation for future learning. Werner et al. (2012) evaluated the products developed by students using fairy tale scenarios in the context of CT skills by using the game programming tool Alice in the introductory programming course at the secondary school level.

A study within the scope of the evaluation of CT skills; difficulties and problems encountered; The lack of consensus on the sub-skill dimensions of the CT skill, the lack of resources and the lack of diversity in the available resources, the lack of automatic assessment tools and professional development seminars (Yeni, 2017).

In the literature review, there is not a clear model and method in the evaluation process of computational thinking skills, and there was no generally accepted scale tool for secondary school students. The measurement tools related to computational thinking skills in Turkish and English languages were examined, and since the dimensions differed considerably in different scales, and there was no determinative scale suitable for the level of secondary school students, it was decided by the researchers to develop a new scale with a clear theoretical basis and views of considered as one of the experienced group teachers.

2. Method

Since the content of the study consists of the scale development process, the survey model was used as a method in the study. The survey model is the scanning arrangements made on a group, sample, or sample taken from the universe in general or taken from it to make a general judgment about the universe in a universe consisting of many elements (Karasar, 2013). In addition, the scale development principles proposed by DeVellis (1991) were also taken as a basis for the development process of the scale in this study. Confirmatory and exploratory factor analysis (CFA and EFA) was used for data analysis in the study. In the study, this model was preferred because it aimed to develop a valid and reliable measurement tool to measure the computational thinking skills of secondary school students.

2.1. Sampling

The scale, which was revised in line with expert opinion, was piloted at the level of four different secondary schools (One private school, and three public schools) in the Milas district. The research application permission to conduct scale forms schools was obtained from Muğla Governorship Provincial Directorate of National Education. Different references were taken in the order determining the study group of the scale. According to Kline (1994), a sample of 200 participants may be sufficient for a factor analysis that does not include more than 40 items.

As the research sample, the 6th and 7th grades which the Problem-Solving and Programming unit had been taught the previous year were determined. In the pilot stage, the scale form was applied to 272 secondary school students, and these data were used in Exploratory Factor Analysis (EFA). Confirmatory Factor Analysis (CFA) was applied on the second stage , a scale form was conducted on a total of different 285 secondary school students. Information details about the sampling are presented in Table 2.

Table 2. Demographic information of research participants

		Stage	Stage
		N=272	N=285
Gender	Female	158	162
	Male	114	123
Age	11	88	85

	12	115	136
	13	69	74
Grade	6 . Grade	174	107
	7. Grade	98	178
Schools	Sakarya Secondary School	126	144
	Çamköy Secondary School	71	65
	Private Özge Secondary School	44	32
	Gümüşlük Secondary School	28	44

3. Results

In this part, findings have been explained according to the scale development principles proposed by DeVellis (1991). During the scale development process, the researcher followed the steps below.

3.1. Application

In the literature review, there is no clear model and method in the process of evaluating computational thinking skills, and there is no generally valid scale tool. The existing measurement tools related to computational thinking skills in Turkish and English languages were examined, and since the dimensions differed considerably in different scales, and there was no determinative scale suitable for the level of secondary school students, the researchers decided to develop a new scale with a clear theoretical basis and views of considered as one of the experienced group teachers.

In the scale development process, the researcher developed the computational thinking (CT) skill self-assessment scale in line with the following principles of DeVellis (1991).

3.2. Determining the operational definition of CT by conducting an in-depth literature review

Although there are many technical aspects in scale development and validity analysis, the importance of grounding the scale well in important theories related to the phenomenon to be measured should not be ignored. The boundaries of the phenomenon should be clearly defined so that the scope of the scale is not dragged into undesirable dimensions. The theory is a great convenience in clarification (DeVellis, 1991). While Papert (1980), a software expert, first introduced CT in 1996, computer scientist Wing (2006) pioneered the concept by further expanding the definition. According to Wing (2014), computational thinking is an area of thinking skills that includes problem-solving, system design, and understanding human behavior by grounding concepts in computer science. According to ISTE -International Science for Technology in Education- (2016), CT is seen as a key need that individuals will need today and tomorrow and is defined as a problem-solving approach that strengthens technology and thinking. According to ISTE, ISTE is a problem-solving process that involves formulating problems using computers and other tools, logically analyzing and organizing data, automating solutions with algorithmic thinking, analyzing, defining, and implementing solutions that are the most efficient and effective combination of steps, and resources.

3.3. Interviewing experienced teachers in the field during the process of creating the item pool

In the creation of the item pool, the opinions of 17 Information Technologies teachers working in the field were taken by using semi-structured interview forms, and it was tried to determine the appropriate items to increase the content validity of the scale. During the interview, questions were asked about the definition and dimensions of computational thinking and the methods used by the teachers toward CT. Content analysis was used in the analysis of the interview forms. In line with the codes obtained as a result of this analysis and literature review, a total of 40-item scale pool was created. These items were first evaluated within the scope of language validity by a Turkish Language and Literature Teacher. All of the items have been developed in a positive structure. The following points were taken into account in the writing of the item pools:

- Attention was paid to the selection of items reflecting the purpose of the scale.
- Long expressions that have multiple meanings, create uncertainty, and examine multiple skills were avoided.
- The technical terms are tried to be simplified as much as possible, they are written, sufficiently short, and by the spelling rules.

3.4. Determining the scale type

The scale was developed in Likert type in terms of structure. In Likert-type scales, there is an action of responding at various levels of agreement or approval to the item or statement presented as a statement expressing a statement. For this reason, the response options were developed at a 5-level consisting of *Strongly disagree (1)*, *Disagree* (2), *Partially agree* (3), *Agree* (4), and *Strongly agree* (5) options with equal intervals according to the difference in the level of agreement.

3.5. Initial item pool reviewed by experts

The content of a scale should validly reflect the conceptual definition (DeVellis, 1991). The draft form created for content and face validity was sent to subject area experts and asked for their evaluation. In this draft, the purpose and scope of the scale's development and the expectations from the experts are clearly stated in the form of a directive. This expert group consisted of one faculty member working in the Curriculum and Instruction department of Universities and three faculty members working in Computer Teaching and Technologies Education. Experts who evaluated the items in the draft form were asked to evaluate the items as 1 (The item is suitable), 2 (Can be improved), and 3 (The item is not suitable). In line with the feedback from the experts, the initial items were re-evaluated and four items were removed from the scale, revised 8 items, and the number of items was reduced to 36.

3.6. Analyze

The focus of scale development is to reveal the structure of the feature that is intended to be measured in the best way. For this purpose, it was carried out to reveal how the predicted structure of the item pool, was prepared and pre-implemented (Erkuş, 2016). According to Jöreskog and Sörbom (1993), factor analysis, which is a common statistical technique applied to reveal this structure and to name these structures, was defined to analyze the mutual relations between variables. Factor analysis has a very important place in measuring psychological structures (Nunnally, 1978).

Before the analysis phase, the missing values in the data set were checked. Missing values can cause serious errors in item load calculations, especially in factor analysis studies, and reduce the statistical power of research (Cole, 1987; Davey & Savla, 2010). According to Tabachnick and Fidell (2007), null values in the research data should not exceed 5%. Each variable in the study data set was examined with frequency tables one by one and it was determined that the null values were not above 5% and the data showed a random distribution. For this reason, no data assignment was made for the missing data.

In many data sets, observations that cannot be controlled, are affected by other variables, and do not belong to the variance can be found. These observations can be more or less. The variables resulting from these observations are called outliers (Hinkle, Wiersma, & Jurs, 1981). The total score and Mahalanobis distance were calculated to determine the extreme values in the data set. Mahalanobis distance is a value used to detect the presence of extreme values that make it difficult to meet linearity and normality assumptions in regression analysis (Çokluk, Şekercioğlu, & Büyüköztürk, 2012). As mentioned by Osborne and Overbay, outliers were examined using z scores for univariate outliers and Mahalanobis distance for multivariate outliers, and the results of the study were more reliable. According to Mahalanobis (D2) distance calculation (1-CDF.CHISQ(MAH_2,6)) for multivariate outliers, 36 outliers were removed from the data set because they were below 0.01.

The meaning of the normality distribution at each factor or each subgroup of the within-group factor of the measurements of the dependent variable is called "multiple normality". Methods and techniques applied for univariate normality can also be used in the examination of multiple normality (Demir et al., 2016; Stevens, 2009; Tabachnick & Fidell, 2007;). By graphical, statistical, and descriptive methods, the position of the scores on a line or whether they show a bell-shaped distribution is determined, and the distribution of the scores is presented visually (Field, 2009; Quinn & Keough, 2002). The Scatter Plot Matrix method was used in this study for the normality distribution and linearity of multivariate data. The shapes of the distributions in this matrix give an idea about normality and linearity. The departure from the ellipse shape in the normality distribution indicates that there is no normal distribution. As the kurtosis and skewness values are in the range of ± 1 the univariate normality assumption was met and the normality distribution and linearity distribution of the multiple variables in the data set are shown in Figure 1. (Mertler & Vannatta, 2005).

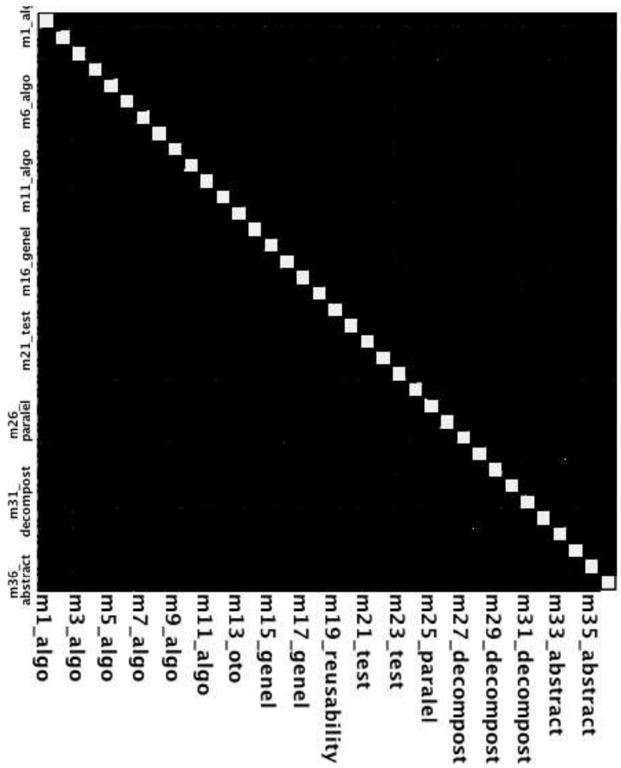

Figure 1. Scatter Plot Matrix

To ensure the construct validity of the scale, exploratory factor analysis, and Confirmatory Factor Analysis techniques were used. There is a widespread consensus in the literature that factor analysis is a statistical technique used to demonstrate construct validity (Anastasi, 1982; Cronbach, 1970; Erkuş, 2016; Reio & Wisell, 2006). In the study, within the scope of the explanatory factor analysis method; principal components analysis was used. Although it is stated that principal component analysis, which is frequently mentioned in the literature (Floyd & Widaman, 1995; Ford, MacCallum, & Tait, 1986; Gorsuch, 1983; Mulaik, 1990; Snook & Gorsuch, 1989), is a different analysis from confirmatory factor analysis, according to some theorists, factor analysis is also seen as an extension of principal component analysis with different assumptions about possible patterns between variables (Arrindell & Van der Ende, 1985; Çokluk et al., 2012; Guadagnoli & Velicer, 1988; Steiger, 1990;). Principal component analysis was defined by Kline (1994) as a method of condensing the correlation matrix. The main difference between these two analyzes is; factor analysis is based on the correlation matrix between the variables, while principal component analysis is based on the variance-covariance matrix. For the difference between the components obtained from this correlation matrix and the factor, Kline (1994) stated that the components are real factors. According to Kline (1994), the general factors of factor analysis are hypothetical. Because these are estimates derived from estimated data.

Before starting the factor analysis, the factorability of the correlation matrix should be tested. Factorization depends on the fact that the relations between variables (scale items) are above a certain level (Pett et al., 2003). The strength of the relationships can be evaluated with Bartlett's Test of Sphericity and the Kaiser-Meyer-Olkin Test of Sampling Adequacy. The fact that the KMO value is .91 in Table 2 indicates that a high-level evaluation can be made for sample adequacy. The Bartlett Test of

Sphericity tests the significance of the difference between the observed correlation matrix (inter-item correlation matrix) and the unit matrix. For the matrix to be factored, the correlations between the items must be statistically significantly different from zero. When the Barlett's Sphericity test results in Table 3. are examined, it is seen that the chi-square ($\chi 2$) value obtained is significant at the 0.01 level and the KMO value.906 was found. In this respect, it was determined that the difference between the matrices was significant to accept the data suitable for analysis and sample adequacy was enough ($p<.05$)

Table 3. Barlett's Test of Sphericity

Kaiser-Meyer-Olkin Sample Test		.906
Barlett's Test of Sphericity	Chi-Square	1684.9
	df	231
	p	.000

The principal component analysis method was used as a factorization method in the study. The main reasons for choosing this method are as follows. This method was preferred to reveal the basic dimensions of the measured subject, to ensure a multivariate normal distribution, to have at least equally spaced data, to have low error variance in the data, to be a scale development study, and to determine under which dimensions the items would be grouped.

According to Tabachnick and Fidell (2007), if the structure is stable and consistent, the result is the same no matter which rotation method is used. Vertical rotation, one of the rotation methods, was used by the researcher in the study. As a result of the analysis, it was seen that the structure was more clearly exposed with vertical rotation. For this reason, varimax technique, one of the vertical rotation methods, was used to extract the items and to see the structure more clearly, and it was assumed that the factors were unrelated to each other. Although the factor loading values vary, the common factor variance gives the same result when calculated over the load values before rotation (Çokluk et al., 2012). It is a model proposed by Kaiser and is a modification of the Quartimax. In this method, which gives priority to the columns of the factor loads matrix while reaching the simple structure, some load values in each column are approached to 1 while the remaining many values are approached to 0.

As a result of the first analysis, it was seen that there were 8 components with an eigenvalue above 1 for the 36 items taken as the basis, and the common variances of the items were calculated. The eigenvalue of a factor reflects the strength of the relationship between the factor and the original variables. For this reason, eigenvalues are used to decide the number of factors.

Instead of directly subtracting items under .40 in the common variance load, factor loads in the rotated components matrix were also considered. Factor load; shows the correlation of the item with the relevant factor. These values are taken into account when determining whether an item is under any factor. It is concluded that items with sufficiently high factor loadings measure a similar structure (Erkuş, 2016). In this study, the .32 value stated by Tabachnick and Fidell (2007) was accepted as the lower limit during the Exploratory Factor Analysis (EFA) stage, and the items to which the loads below this value belong were not taken into account. According to Erkuş (2016); It was stated that in reducing the items, especially in studies with a low number of items, items remove according to this rule caused the structure to deteriorate. For this reason, by using different rotation and subtraction techniques in the study, the incorrect creation of the structure was prevented. Considering the reviewed by experts state (36 items) of the scale as a result of the item loads in Table 3, 16 items (I1, I3, I6, I9, I10, I11, I15, I17, I18, I19, I21, I22, I29, I31, I33, and I36) were removed. In the item removal process, the overlap between the items and the factors were taken into consideration. substance overlap status; the item giving a load value above the acceptance level (.32) in more than one factor and the difference between the load values of the item in two or more factors is less than .10 (Çokluk et al., 2012). As a result of the analysis, the total number of these components decreased to 5 and the explained variance rate was 55.165%. According to Scherer et al. (1988), a variance explanation rate of between 40% and 60% is considered sufficient for studies in the field of social sciences. The high rate of variance explained is as much as the strength of the factor structure of the scale (Gorsuch, 1983). The ratio of explained variances is 32%, 6.8%, 5.7%, 5.4%, and 5.1%.

Table 3. Explained Total Variance

Components	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings		
	Total	% of Variance	Cum. %	Total	% of Variance	Cum. %	Total	% of Variance	Cum. %
1	6.433	32.167	32.167	6.433	32.167	32.167	2.856	14.282	14.282
2	1.337	6.687	38.854	1.337	6.687	38.854	2.580	12.900	27.182
3	1.154	5.771	44.625	1.154	5.771	44.625	**2.314**	11.568	38.750
4	1.088	5.438	50.063	1.088	5.438	50.063	1.697	8.487	47.238
5	1.020	5.102	55.165	1.020	5.102	55.165	1.585	7.927	55.165
6	.924	4.622	59.787						
7	.822	4.111	63.898						
8	.805	4.025	67.923						
9	.776	3.881	71.803						
10	.720	3.602	75.405						
11	.665	3.323	78.728						
12	.643	3.213	81.942						
13	.594	2.970	84.911						
14	.554	2.768	87.679						
15	.496	2.481	90.160						
16	.448	2.242	92.402						
17	.424	2.119	94.522						
18	.404	2.019	96.541						
19	.373	1.867	98.408						
20	.318	1.592	100.000						

Extraction Method: Principal Component Analysis.

The values in the common variance loads (communality) field shown in Table 4 are the correlation value of the factors. This value is high for .6 and above, the average for .6 and .3, and unacceptable for below .3 (Kline, 1994).

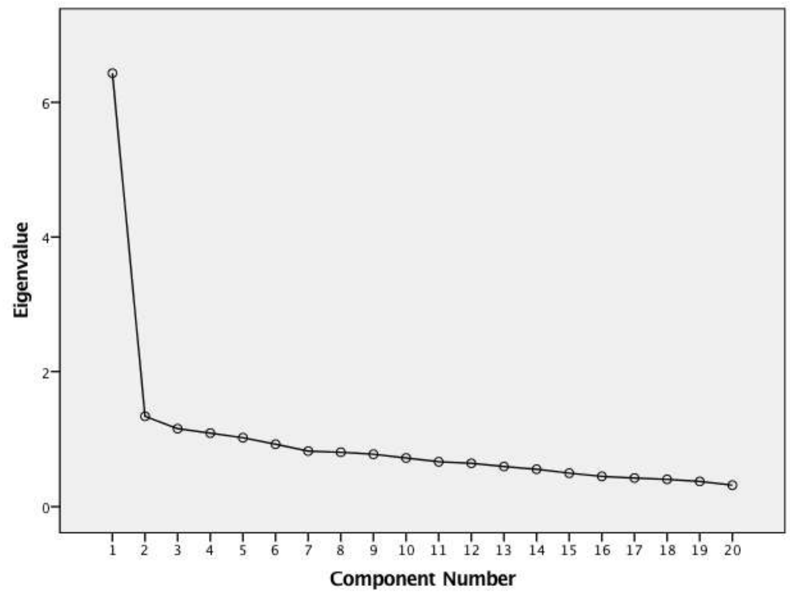

Figure 2. Scree Plot Graph

When Figure 1 is examined, 5 factors can be selected according to the eigenvalue based breaking point. However, this is a visually decided interpretation. Deciding according to the Total Variance Explained in Table 4. will help us reach a more accurate result.

Table 4. Exploratory Factor Analysis Results

		Factor Loads				
	h²	Factor1	Factor 2	Factor 3	Factor 4	Factor 5
I4 - I can determine the steps I need to follow in solving the problem myself.	.53	.67				
I6 - I can develop the algorithms I use in my applications myself.	.56	.64				
I5 - I set the limits of the loops I use in my applications.	.49	.62				
I20 - I make changes to a working code block.	.55	.56				
I7 - I calculate which processes should be within loop boundaries in my applications.	.48	.56				
I23 - I easily find the errors I encounter in my applications.	.60		.73			
I34 - I use a programming language to solve problems.	.46		.60			
I25- I use multiple events in my applications.	.60		.59			
I24 - I use multiple objects in the same scene in my applications.	.60		.56			
I26 - I describe events that happen in the background while my apps are running.	.54		.54			

I8 – In problem-solving, I produce solutions using operators (+, -,*, /), expressions, and equations.	.52		.67		
I27 - I remove unnecessary steps in my application.	.63		.65		
I14 - I can show the solution to problems with formulas.	.62	.	.64		
I28- I organize the process steps in order of importance in my applications.	.53		.61		
I35- I do not feel obligated to use a programming language to solve problems.	.69			.80	
I30- I break down the problem-solving process into smaller parts.	.49			.57	
I3- I can classify the problems I encounter as easy, medium, and difficult.	.61			.54	
I2- I think that I use the shortest way to solve the problem.	.55				.70
I13- I can have computers and other tools (Robot, tablet, phone, etc.) perform repetitive and similar operations.	.48				.54
I12- I use the computer or other tools to solve problems.	.50				.52

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.

In addition, GFI=.910, CFI=.944, RFI=.805, IFI=.946, NFI=.936, RMSEA=.043, AGFI=.889, and PGFI=.898 values, optimal values were obtained for the fit of the model as seen in Table 5. Improvement on the model was not required. (Byrne, 2010;Raykov & Marcoulides, 2006). GFI values between .90 and 95 are considered acceptable model indicators (Byrne, 2010). RMSEA value of less than 0.05 is considered to be a good fit, less than .08 is acceptable, and a value above is considered to indicate a mediocre fit (Brown, 2006; Hoe, 2008; Jöreskog & Sörbom, 1993).

Table 5. Fit Values of model

Index	Scale Value	Perfect Fit	Good Fit
X^2 /sd	2.6	X^2 /sd ≤ 2	X^2 /sd ≤ 3
CMIN/DF	1.415	CMIN/DF<2.5	CMIN/DF <3
RMSEA	.043	0<RMSEA<.05	0,05<RMSEA≤.1.0
CFI	.95	.97<CFI<1	.95<CFI<.97
GFI	.92	.95<GFI<1	.90<GFI<.95
RFI	.805	.90<RFI<1	.65<RFI<.89
NFI	.936	.95<NFI<1	0,90≤NFI≤0,95
IFI	.95	.80<IFI<.85	.85< IFI< 1.0
AGFI	.889	.90<AGFI<1	0,85≤AGFI≤0,90
PGFI	.898	.95<PGFI<1	0,85≤PGFI≤0,90

CFI: Comparative Fit Index, GFI : Goodness of Fit Index, RFI: Relative Fit Index, RMSEA: Root Mean Square Error of Approximation, IFI: Incremental Fit Index, PGFI: Parsimony Goodness of Fit Index

The correlation between the factors of the five-factor model is given in Table 6. Accordingly, it is seen that there is no clear consensus among the sources in determining the sub-dimensions of the CT field in the literature. According to the results obtained by calculating the Pearson correlation values, it was observed that there was a correlation of .80 and .82 between algorithmic thinking and respectively parallelization and automatization factors.

Table 6. Pearson Correlation Values Between Factors and Cronbach's Alpha (α) Values

	α	1	2	3	4	5
Algorithmic Thinking (1)	.81					
Abstraction (2)	.83	.64				
Parallelization (3)	.88	.82	.75			
Automation (4)	.85	.80	.59	.70		
Decomposition (5)	.90	.75	.77	.77	.62	

*p<.05, α = Cronbach Alpha Value

According to Kline (1994), in the measurement model, the correlation estimates between the factors in the CFA results and the loads of the factors on which the indicators depend should be specified. Cronbach Alpha values show that all factors have high reliability In addition, if the model is validated logically, as in this study, it should be taken into account that the correlation estimates between the factors should not be too high (>.85) (Çokluk et al., 2012). It was determined that there was no correlation above .85 in the model, and the lowest correlation was .59 between the abstraction and automatization factors.

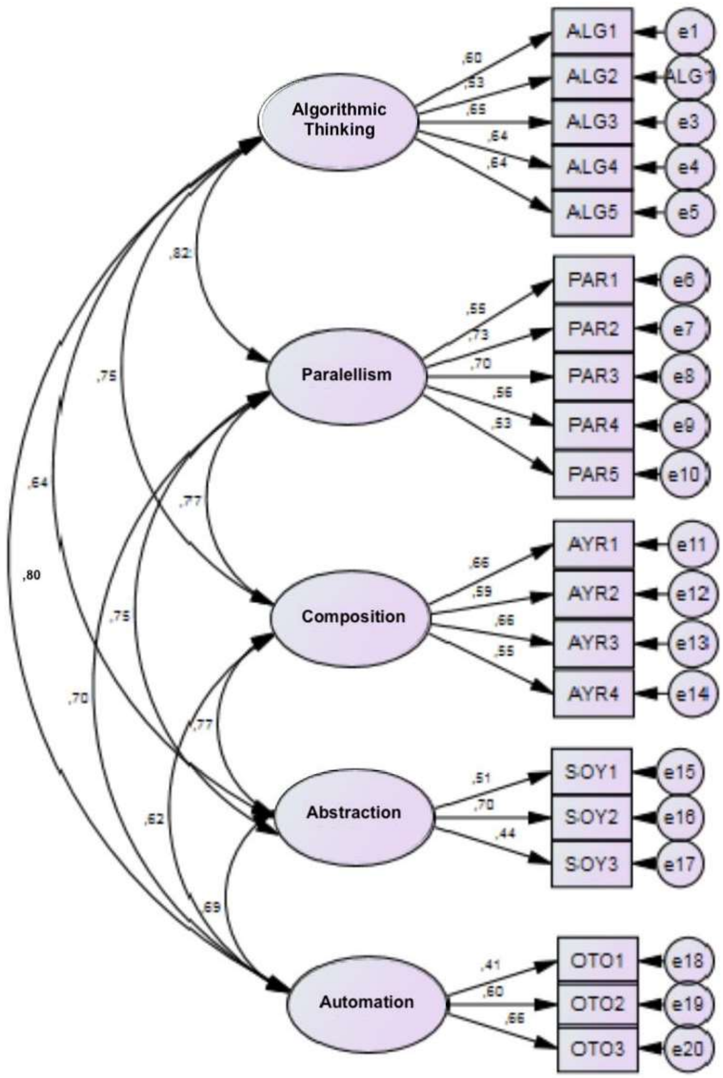

Figure 2. Confirmatory Factor Analysis Model of the Scale.

Confirmatory factor analysis (CFA) is an advanced technique based on testing theories with latent variables (Tabachnick & Fidell, 2007). This technique is also used to validate structures or models. IBM Amos v22 program was used for CFA in the study. CFA, in which the model developed as a result of EFA is also seen as a hypothesis test, is widely used to show the relationships between the observed variables and latent variables and among themselves in the latent variables. Figure 2 shows the setup of the theoretical structure in the AMOS environment.

As a result of the alpha test performed for the reliability analysis of the scale, Cronbach's alpha value was obtained as .89 as indicated in the Table 5. This value is sufficient for the reliability level of the scale.

In addition, in the reliability analysis of the factors, the Cronbach alpha value of the algorithmic thinking factor was .75, the decomposition factor was .71, the parallelization factor was .75, the abstraction factor was .58, and the automation factor was .56. As a result of EFA and CFA, a total of 20 items remained in the scale.

Discussion and conclusion

Bocconi et al. (2016) stated in the report they presented to the European Union Education Commission that the CT skills that will be gained through coding and programming education at the basic education level will have two basic gains for both individuals and countries. In the first of these gains, CT makes a significant contribution to economic growth, fills the unemployment in the field of information and communication technologies, and prepares for the occupational groups that will emerge in the future, and as the second gain, the development of CT skills in individuals provides them with different thinking skills and the ability to express themselves using various tools. stated that it is the acquisition of problem-solving skills and solving daily issues from different viewpoints. In this context, it is of great importance that CT is taught at the basic education level. Determining at what level this skill is acquired during the education program implementation process is seen as an important element that will increase the quality of teaching this skill.

In this study, a multi-dimensional scale consisting of five sub-factors and 20 items was developed to measure the computational thinking skills of secondary school students 11-13 age group in Turkey. In the writing of the scale items, negative sentence structures were not used frequently to avoid confusion of meaning among the participants. But there is one item on the abstraction factor with negative statements, this item should be reverse-coded when coding the responses on the scale. The highest score that can be obtained on the scale is 100, while the lowest score is 20. The high score obtained indicates that the participants' self-evaluation of CT skills is high.

Result of the study; it was seen that the scale had sufficient psychometric properties. It was developed that reveals the self-assessment of computational thinking skills by enabling students to question themselves better, and the necessary reliability and validity analyzes were made to meet the expectations. Although there are different understandings about CT sub-dimensions of CT (Kalelioğlu et al,2014; Barr & Stephenson, 2011; Grover et al., 2013; Selby & Woollard, 2013). In the literature review, it is seen that there is not a single generally accepted component about the components of CT. The concepts and processes in the computational thinking process can be listed as abstraction, algorithm design, automation, data collection, data analysis, data presentation, decomposition, simultaneous operation, pattern recognition, pattern generalization, and modeling. In this scale, the sub-dimensions mentioned in the literature were also reached. However, according to different views accepted in the literature (ISTE, 2016; Kalelioğlu et al., 2014; Selby & Woollard, 2013; Wing, 2006), these components are; abstraction, algorithmic thinking, evaluation, problem-solving, decomposition, pattern recognition, and generalization are important components of the CT.

As a result of the analyses of the study, a valid and reliable scale consisting of five factors with twenty items was obtained. In this study; it has been determined that computational thinking skill consists of five factors as seen in Figure 3. Among the factors of the scale, there are five items of algorithmic thinking, five items of parallelization, four items of decomposition, three items of automation, and three items of abstraction. Cronbach Alpha value determined that there was no correlation above .85 in the model, and the lowest correlation was .59 between the abstraction and automatization factors. The scale explains 55.17% of the total variance. There is no exact value for the minimum variance that a scale should explain, but it is stated that a variance explanation rate of between 40% and 60% is considered sufficient for studies in the field of social sciences (Scherer et al.,1988).

In the second stage, as a result of the confirmatory factor analysis of the five-dimensional and 20-item, good fit values were estimated, and thus construct validity was ensured. Although there is no consensus in the literature about the model fit indices to be considered in determining the model fit in CFA, in addition to the χ^2/df value (Kline, 1994), RMSEA (Jöreskog & Sörbom, 1993), CFI (Bentler, 1990) and SRMR (Hoe, 2008; Brown, 2006) fit indices are frequently recommended. In addition, GFI, RFI, IFI, NFI, AGFI, and PGFI values, optimal values were obtained for the fit of the model. Improvement on the model was not required (Byrne, 2010; Raykov & Marcoulides, 2006). For this reason, these fit indices were used in model evaluation in the research. As a result of the analysis of the square root of AVE, CR, and AVE and the correlation coefficients between the factors reached by CFA, it was seen that the scale met the conditions of validity. Therefore, it is possible to say that the scale has a stable structure, and consistent results can be met when applied at different sampling. In the reliability analysis of the scale, Cronbach's alpha coefficient was tested and examined in terms of the overall scale and factors. In the reliability analysis of the factors, the Cronbach alpha value of the algorithmic thinking factor was .75, the decomposition factor was .71, the parallelization factor was .75, the abstraction factor was .58, and the automation factor was .56. These values indicate that adequate validity is achieved (Cronbach & Shavelson, 2004; Barbera et al, 2021).

When the validity and reliability indicators of the scale are evaluated together, it can be stated that the scale can be used safely to determine the secondary school students' self-assessment of CT skills.

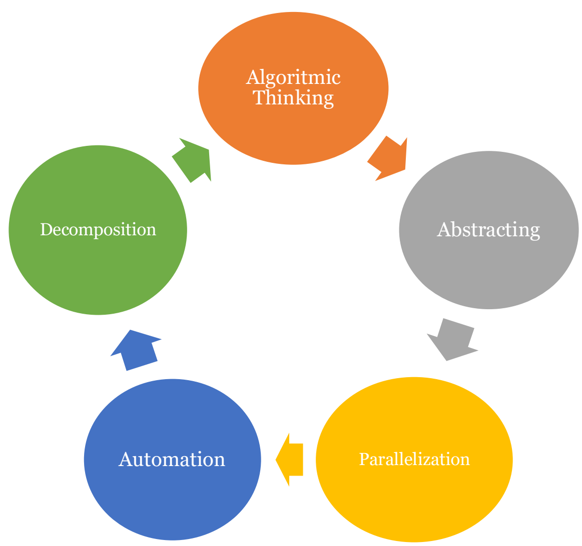

Figure 3. Scale Factors

This developed scale can be used to measure the self-assessment perceptions of students in secondary school students regarding their computational thinking skills. It is thought that this scale can contribute to the practitioners in better directing those who plan to receive education in the field of programming and coding. However, as stated before, since the computational thinking skill is an abstract skill, based it on a single measurement tool may not give very accurate results. Therefore, it is recommended to use different assessment tools together (Gouws, Bradshaw & Wentworth, 2013; Özmen & Altun, 2016; Yeni, 2017). The final version of the Turkish version with all items of the scale developed in this study is presented in Appendix 1.

In recent years, there is a need for research on computational thinking skills, which have been tried to be gained by different methods at different ages, at which ages, with which tools, environments, and methods, more effectively and permanently (Gülbahar et al., 2019; Lockwood & Mooney, 2017). The limitation of the research is that it was conducted in the 2018-2019 academic year with the students attending the sixth grades of one private and three public secondary schools in the Milas district of Muğla province.

References

Ambrosio, A. P., Almeida, L. S., Macedo, J., & Franco, A. (2014). Exploring core cognitive skills of computational thinking. *Psychology of programming interest group*. Annual conference 2014 (PPIG 2014) (ss. 25–34).

Anastasi, A. (1982). *Psychological Testing*. (6th ed.). New York: McMillan.

Armoni, M., Meerbaum-Salant, O., & Ben-Ari, M. (2015). From scratch to “real” programming. *ACM Transactions on Computing Education*, 14, 4.

Arrindell, W. A., & Van der Ende, J. (1985). An empirical test of the utility of the observer-to-variables ratio in factor and components analysis. *Applied Psychological Measurement*, 9, 165–178.

Barbera, J., Naibert, N., Komperda, R., & Pentecost, T. C. (2021). Clarity on Cronbach’s Alpha use. *Journal of Chemical Education*. 98, 257-258. DOI: 10.1021/acs.jchemed.0c00183

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, 48-54.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education-implications for policy and practice. Seville: *Join Research Center (European Commission*). 21.02.2018, http://publications.jrc.ec.europa.eu/repository/bitstream/JRC104188/jrc104188_computhinkreport.pdf.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *In Annual American Educational Research Association Meeting*. British Columbia: Vancouver.

Cole, D. A. (1987). Utility of confirmatory factor analysis in test validation research. *Journal of Consulting and Clinical Psychology,* 55, 584–594. [6]

Cronbach, L. J. (1970). *Essentials of Psychological Testing*. (3rd ed.). New York: Harper and Row Pub.

Cronbach, L. J., & Shavelson, R. J. (2004). My Current Thoughts on Coefficient Alpha and Successor Procedures. Educucational Psychology Measurement, 64 (3), 391–418.

Cuny, J., Snyder, L., & Wing, J. (2010). *Demystifying computational thinking for non-computer scientists*. 12.12.2018, http://people.cs.vt.edu/~kafura/ CS6604/Papers/CT-What-And-Why.pdf

Davey, A., & Savla, J. (2010*). Statistical Power Analysis With Missing Data. A Structural Equation Modeling Approach*. New York: Taylor & Francis Group, LLC.

Denning, P. J. (2007). Computing is a natural science. *Commun. ACM,* 50, 13-18.

Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational Thinking A Guide for Teachers. 02.01.2019, http://eprints.soton.ac.uk/id/eprint/424545

Çokluk, Ö. S., Şekercioğlu, G., & Büyüköztürk, S. (2012). *Sosyal Bilimler İçin Çok Değişkenli İstatistik: Spss ve Lisrel Uygulamaları*. Ankara: Pegem Akademi Yayıncılık.

Demir, E., Saatçioğlu, Ö., & İmrol, F. (2016). Uluslararası Dergilerde Yayımlanan Eğitim Araştırmalarının Normallik Varsayımları Açısından İncelenmesi. *Curr Res Educ*. 2(3). 130-148

DeVellis, R. F.(1991). *Scale Development: Theory and Applications*. Newbury Park, CA: Sage; 1991.

Edelson, D. (2002). *Pedagogical Aspects of Computational Thinking*. 12.12.2018. Available: http://www.nap.edu/catalog.php?record_id=13170

Erkuş, A. (2016). *Psikolojide Ölçme ve Ölçek Geliştirme I: Temel Kavramlar ve İşlemler*. Pegem akademi. (3. Baskı). Ankara

Field, A. (2009). *Discovering Statistics Using SPSS*. London: SAGE Publications.

Floyd, F. J., & Widaman, K.F. (1995). Factor Analysis in The Development and Refinement of Clinical Assessment Instruments. *Psychological Assessment, 7(*1), 286–299.

Ford, J. K., MacCullum, R. C., & Tait, M. (1986). The Application of Exploratory Factor Analysis in Applied Psychology: A Critical Review And Analysis. *Personnel Psychology,* 39, 291–314.

Garcı̄a-peñalvo, F. J., Rees, A. M., Hughes, J., & Vermeersch, J. (2016). A survey of resources for introducing coding into schools. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality* (TEEM'16), 19–26.

Gorsuch, R.L. (1983). *Factor analysis* (2nd ed.). Lawrence Erlbaum Associates, Inc.. ISBN-10: 089859202X

Gouws, L., Bradshaw, K., & Wentworth, P. (2013). First year student performance in a test for computational thinking. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference on - SAICSIT'13*. doi:10.1145/2513456.2513484

Grover, S., Cooper, L., & Pea, R. (2013). Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43.

Guadagnoli, E., & Velicer, W.F.(1988). The releationship of the sample size to the stability of component patternes. *Psychological Bulletin,* 103, 265-275.

Guzdial, M., Kay, A., Norris, C., & Soloway, E. (2019). Computational Thinking Should Be Good Thinking. *Communications of the ACM*, November 2019, Vol. 62 No. 11, Pages 28-3010.1145/3363181

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi İşlemsel Düşünme Becerisine Yönelik Öz Yeterlik Algısı Ölçeği: Geçerlik ve Güvenirlik Çalışması . *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* , 10 (1) , 1-29 . DOI: 10.16949/turkbilmat.385097

Hinkle, D.E., Wiersma, W., & Jurs., S. G. (1981). *Applied Statistics For The Behavioral Sciences*. Chicago: Rand McNally.

ISTE (2016). Computational Thinking (Learner). *ISTE website, Computational Thinking,* date of access: 12.08.2017. https://iste.org/standards/computational-thinking

Jöreskog, K. G., & Sörbom, D. (1993). *LISREL 8: Structural Equation Modeling With The SIMPLIS Command Language*. New York: Scientific Software International, Inc.

Karasar, N. (2013). *Bilimsel Araştırma Yöntemi*. İstanbul: Nobel Yayin Dağitim .

Kalelioğlu, F., & Gülbahar, Y. (2015). Bilgi İşlemsel Düşünme Nedir ve Nasıl öğretilir?.*3th International Instructional Technology and Teacher Education Symposium*. Trabzon, Türkiye, September 9 – 11, 2015.

Kalelioğlu, F., Gülbahar, Y., Akçay, S., & Doğan, D. (2014). Curriculum Integration Ideas for Improving the Computational Thinking Skills of Learners through Programming via Scratch. *The 7th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*. Istanbul University, Turkey, September 22-25.

Kline, P. (1994). *An Easy Guide To Factor Analysis*. Routledge Press.

Lockheed, M. E. (1985). Women, girls, and computers: A first look at the evidence. *Sex Roles*, 13, 115–122. Access date: 12.11.2018, https://link.springer.com/article /10.1007/ BF00287904.

MacCallum, R. C., & Tucker, L.R. (1991). Representing sources of error in the common factor model: Implications for theory and practice. *Psychological Bulletin,* 109(3), 502.

Mertler, C.A., & Vannatta, R.A. (2005). *Advanced And Multivariate Statistical Methods: Practical Application And Interpretation. (Third edition).* United States: Pyrczak Publishing.

Mulaik, S.A. (1990). Blurring the Distinctions Between Component Analysis and Common Factor Analysis. *Multivariate Behavioral Research*, 25(1), 53-59.

Nunnally, J.C. (1978). Psychometric theory (2nd ed.). New York: McGraw-Hill

Orton, K., Weintrop, D., Beheshti, E., Horn, M., Jona, K., & Wilensky, U. (2016). Bringing computational thinking into high school mathematics and science classrooms. *Transforming Learning, Empowering Learners*. The International Conference of The Learning Sciences (ICLS) (ss. 705–712).

Özmen, B., & Altun, A. (2014). Undergraduate students' experiences in programming Difficuities and obstacles. *Turkish Online Journal of Qualitative Inquiry*, July 2014, 5(3)

Papert, S. (1980). *Computers For Children. Mindstorms: Children, Computers, and Powerful Ideas.* Basic Books, Inc. Publishers / New York

Pett, M.A., Lackey, N.R., & Sullivan, J.J. (2003) *Making Sense of Factor Analysis: The Use of Factor Analysis for Instrument Development in Health Care Research*. SAGE Publications, Thousand Oaks.http://dx.doi.org/10.4135/9781412984898

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In Proceedings of the 41st *ACM Technical Symposium On Computer Science Education (SIGCSE* '10) (pp. 265–269). New York, NY: ACM Press.

Scherer, R.F., Wiebe, F.A., Luther, D.C., & Adams J.S. (1988), Dimensionality of Coping: Factor Stability Using the Ways of Coping Questionnaire. *Psychological Reports*, 62, 763-770.

Schwartz, D.L., & Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction*, 22(2), 129–184.

Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *In Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 59-66. ACM

Selby, C., & Woollard, J. (2013). *CT: The developing definition.* Erişim Tarihi:14.11.2018 http://eprints.soton.ac.uk/356481/.

Snook, S. C., & Gorsuch, R.L. (1989). Common factor analysis vs. component analysis. *Psychological Bulletin,* 106,148–154.

Steiger, J.H. (1990). Some Additional Thoughts on Components, Factors, and Factor Indeterminacy. *Multivariate Behavioral Research*, 25(1), 41-45.

Stevens, J. P. (2009). *Applied Multivariate Statistics For The Social Sciences (Fifth edition).* United States: Taylor and Francis Group, LLC.

Tabachnick, B. G., & Fidell, L. S. (2007). *Using Multivariate Statistics* (5. Baskı). Boston: Allyn and Bacon.

Quinn, G. P., & Keough, M., J. (2002). *Experimental Design and Data Analysis For Biologists*. United States: Cambridge University Press

Werner, L., Campe, S., & Denner, J. (2012). Children learn computer science concepts via Alice game programming. *In Proceedings of the 43rd ACM Technical Symposium On Computer Science Education (SIGCSE '12)* (pp. 427–432). New York, NY: ACM.

Wing, J. M. (2006). Computational Thinking. *Communications of the ACM, 49, 33–36*.

Wing, J.M. (2014). Computational Thinking Benefits Society. *40th Anniversary Blog of Social Issues in Computing*, 2014.

Yeni, Z. (2017). Bilgi İşlemsel Düşünmede Değerlendirme. Y. Gülbahar (Editör). *Bilgi İşlemsel Düşünmeden Programlamaya* (341-356). Ankara. Pegem Akademi

Yıldız Durak, H., & Sarıtepeci, M. (2018). Analysis of the Relation Between Computational Thinking Skills and Various Variables with the Structural Equation Model. *Computers & Education*, 116 (2018), 191-202.

Adres
Rumeli Eğitim Araştırmaları Dergisi
İstanbul Medeniyet Üniversitesi, Eğitim Bilimleri Fakültesi, Türkçe ve Sosyal Bilimler Bölümü, Tütrkçe Eğitimi ABD
Cevizli, Kartal - İSTANBUL / TÜRKİYE 34865
e-posta: rumelieducation@yandex.com
tel: +90 505 7958124, +90 216 773 0 616

Address
Rumeli Journal of Education Studies
İstanbul Medeniyet University, Faculty of Education Sciences, Dep. of Turkish and Social Sciences Education, Turkish Education Program
Cevizli, Kartal - İSTANBUL / TÜRKİYE 34865
e-mail: rumelieducation@yandex.com
phone: +90 505 7958124, +90 216 773 0 616

Appendix 1. Self-Assessment of Computational Thinking Skill Scale (Turkish Version)

Sevgili Öğrenciler, bu ölçek sizlerin bilgi işlemsel düşünme becerilerinizin özdeğerlendirmesine yönelik bilimsel bir çalışmada kullanılmak üzere tasarlanmıştır. Lütfen her bir maddeyi özenle ve dikkatle okuyunuz ve kendinize en uygun cevabı "tamamen katılıyorum", "katılıyorum", "kısmen katılıyorum", "katılmıyorum", "hiç katılmıyorum" şeklinde kodlayınız. Katılımınızdan dolayı şimdiden çok teşekkür ederim.

Cinsiyetiniz : Kız ( ) Erkek ( )

Sınıfınız : 6. Sınıf ( ). 7. Sınıf ( )

Günlük olarak Ekran (Akıllı Telefon, Tablet, Bilgisayar, Oyun konsolu vb diğer araçları) Kullanım süreniz:
0-1 saat ( ) 1-2 saat () 2-3 saat ( ) 3-4 Saat ( ) 4 saat ve daha fazlası ( )

		Kesinlikle Katılmıyorum	Katılmıyorum	Kısmen Katılıyorum	Katılıyorum	Kesinlikle Katılıyorum
		(1)	(2)	(3)	(4)	(5)
1	Problemin çözümünde takip etmem gereken adımları kendim belirlerim.	(O)	(O)	(O)	(O)	(O)
2	Uygulamalarımda kullandığım döngülerin sınırlarını belirlerim.	(O)	(O)	(O)	(O)	(O)
3	Uygulamalarımda hangi işlemlerin döngü sınırları içerisinde olması gerektiğini hesaplarım.	(O)	(O)	(O)	(O)	(O)
4	Bir problemin çözümünden yola çıkarak yeni bir problemi çözerim.	(O)	(O)	(O)	(O)	(O)
5	Çalışan bir kod bloğunda değişiklikler yaparım.	(O)	(O)	(O)	(O)	(O)
6	Uygulamalarımda karşılaştığım hataları kolaylıkla bulurum.	(O)	(O)	(O)	(O)	(O)
7	Uygulamalarımda birden fazla nesneyi aynı sahnede kullanırım.	(O)	(O)	(O)	(O)	(O)
8	Uygulamalarımda birden fazla olay kullanırım.	(O)	(O)	(O)	(O)	(O)
9	Uygulamalarım çalıştığı anda arka planda gerçekleşen olayları (işlemleri) açıklarım.	(O)	(O)	(O)	(O)	(O)
10	Problemlerin çözümü için farklı programlama dilleri kullanırım.	(O)	(O)	(O)	(O)	(O)
11	Uygulamamdaki gereksiz (önemsiz) adımları çıkartırım.	(O)	(O)	(O)	(O)	(O)
12	Uygulamalarımdaki işlem adımlarını önem sırasına göre sıralarım.	(O)	(O)	(O)	(O)	(O)
13	Problemlerin çözümünü formül ile gösterebilirim.	(O)	(O)	(O)	(O)	(O)
14	Problemlerin çözümünde operator (+, -,*, /), ifade ve eşitlikleri kullanarak çözüm üretirim.	(O)	(O)	(O)	(O)	(O)
15	Problemin çözüm sürecini daha küçük parçalara ayırırım.	(O)	(O)	(O)	(O)	(O)
16	Problemlerin çözümünde programlama dili ya da programlama kavramı kullanmadan da çözümü gerçekleştirebilirim.	(O)	(O)	(O)	(O)	(O)
17*	Problemlerin çözümü için programlama dili kullanma zorunluluğu hissetmem.	(O)	(O)	(O)	(O)	(O)
18	Problem çözümünde en kısa çözüm yolunu kullandığımı düşünürüm.	(O)	(O)	(O)	(O)	(O)
19	Problem çözümünde bilgisayar veya diğer araçları kullanırım.	(O)	(O)	(O)	(O)	(O)
20	Tekrar eden ve birbirine benzer işlemleri bilgisayarlara ve diğer araçlara (Robot, tablet, telefon vb.) yaptırabilirim.	(O)	(O)	(O)	(O)	(O)

Not: * ters kodlu madde

Adres
Rumeli Eğitim Araştırmaları Dergisi
İstanbul Medeniyet Üniversitesi, Eğitim Bilimleri Fakültesi, Türkçe ve Sosyal Bilimler Bölümü, Tütrkçe Eğitimi ABD
Cevizli, Kartal - İSTANBUL / TÜRKİYE 34865
e-posta: rumelieducation@yandex.com
tel: +90 505 7958124, +90 216 773 0 616

Address
Rumeli Journal of Education Studies
İstanbul Medeniyet University, Faculty of Education Sciences, Dep. of Turkish and Social Sciences Education, Turkish Education Program
Cevizli, Kartal - İSTANBUL / TÜRKİYE 34865
e-mail: rumelieducation@yandex.com
phone: +90 505 7958124, +90 216 773 0 616