

# The Use of Delphi Technique in Validating a Teaching and Learning Model for Enhancing Students' Computational Thinking Skills

Nor Hasbiah Ubaidullah<sup>1,\*</sup>, Zulkifley Mohamed<sup>2</sup>, Jamilah Hamid<sup>1</sup>, Suliana Sulaiman<sup>1</sup>

<sup>1</sup>Department of Computing, Universiti Pendidikan Sultan Idris, Tg. Malim, 35900, Perak, Malaysia

<sup>2</sup>Department of Mathematics, Universiti Pendidikan Sultan Idris, Tg. Malim, 35900, Perak, Malaysia

Received October 18, 2020; Revised November 23, 2020; Accepted December 20, 2020

## Cite This Paper in the following Citation Styles

(a): [1] Nor Hasbiah Ubaidullah, Zulkifley Mohamed, Jamilah Hamid, Suliana Sulaiman, "The Use of Delphi Technique in Validating a Teaching and Learning Model for Enhancing Students' Computational Thinking Skills," *Universal Journal of Educational Research*, Vol. 8, No. 12B, pp. 8201-8213, 2020. DOI: 10.13189/ujer.2020.082624.

(b): Nor Hasbiah Ubaidullah, Zulkifley Mohamed, Jamilah Hamid, Suliana Sulaiman (2020). *The Use of Delphi Technique in Validating a Teaching and Learning Model for Enhancing Students' Computational Thinking Skills*. *Universal Journal of Educational Research*, 8(12B), 3310 - 3322. DOI: 10.13189/ujer.2020.082624.

Copyright©2020 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

**Abstract** Computational thinking skill is a new skill needed in twenty-first century learning. However, teaching practitioners have been struggling to find effective techniques to be used in the teaching and learning process that can help students to develop such a skill. Several studies have proposed a number of models to guide the teaching and learning process, but most lacked a focus on the improvement of computational thinking skill among students. As such, this study was carried out to identify and validate the elements and sub-elements of an assessment instrument prior to developing an effective teaching and learning model for enhancing students' computational thinking skill with the use of the Delphi technique, which was carried out through three rounds of implementation involving a panel of experts made up of five (5) academicians and two (2) industrial practitioners. Essentially, the proposed elements and sub-elements were derived from those based on the combination of the problem-solving techniques and metacognitive skills. After the completion of the three rounds of such technique, six (6) elements and 25 sub-elements of the assessment instrument were validated to be highly important, entailing them to be embedded in the proposed teaching and learning model for enhancing students' computational thinking skills. The use of Delphi technique in this study successfully identified and validated six (6) elements and 25 sub-elements of the teaching and learning model. This study contributes to the

development of a model that can be used by teaching practitioners as a guide in their teaching and learning process for enhancing students' computational thinking.

**Keywords** Computational Thinking Skill, Computer Programming, Delphi Technique, Teaching and Learning Model

---

## 1. Introduction

Essentially, computational thinking skill is a skill that helps an individual to systematically and logically understand the occurrence of an event. According to [1], computational thinking is defined as a way of thinking that can be expressed algorithmically that enables humans in solving problems with the use of specific devices. On the other hand, [2] asserts that problem-solving is a concept of computational thinking, which, according to [1], can be enhanced through learning activities that train specific thinking to enable students use computers more efficiently. However, this skill encompasses not only computing skill but also basic skill that every individual need to possess to help produce digital makers. Admittedly, the literature is replete with many definitions of computational thinking constituting a list of 66 elements, with five topping the list, namely abstraction,

algorithm, decomposition, evaluation, and generalization [3]. Taking cognizance of its importance in the twenty-first century, many nations have started to embed this skill in their school curricula encompassing all levels of education, from elementary to tertiary level. As such, teaching practitioners must use suitable learning and teaching techniques and strategies to help develop such a skill among their students. In this regard, appropriate learning models can be used to guide teachers and lecturers to identify relevant techniques for their classes. Even though many studies of computational thinking skills have been carried out, most lacked a focus on the development of appropriate models for effective learning and teaching that could help improve students' computational thinking skills.

As such, there is an urgent need to develop effective learning and teaching models by taking into account appropriate techniques that can help students develop and improve their computational thinking skills. In the field of computer science, such an important skill is highly relevant to computer programming, which has been demonstrated in several studies of the teaching of computer programming [4-6], the findings of which led to the development of several popular learning and teaching models and techniques to help improve students' computational thinking skills. In this study, the researchers analyzed five leading models to determine their effectiveness in improving students' computational thinking skills, which are closely related to computer programming and problem-solving [7-11]. The analysis helped reveal the essential elements and sub-elements of the five models, which formed the basis of this investigation. Also, several researchers [12-16] have used the metacognitive skill technique in the teaching and learning of computer programming. However, existing teaching and learning models have not used such a technique in teaching the subject matter. The analysis showed a combination of appropriate elements and sub-elements of the five models, problem-solving techniques, and metacognitive skill techniques would be effective in improving students' computational thinking skills.

As such, a test was entailed to verify the validity of elements and sub-elements that had been selected for the development of a learning and teaching model to guide the development of strong computational thinking skills among students. In this respect, the researchers used the Delphi technique to validate the elements and sub-elements of a proposed model. As revealed in many qualitative studies, this technique has been proven to be highly reliable to help practitioners to make informed decisions and reach important consensus in various fields, including education [17]. In principle, this technique operates by searching an agreement or a consensus with the use of assessment instruments or questionnaires that are administered for several rounds to gather data from a panel of experts. In this regard, [18] argues that the Delphi

technique is used to filter and identify relevant factors and to help experts reach an agreement in a particular study. As such, this study used this technique to validate the selected elements and sub-elements of the proposed teaching and learning model.

Premised in this context, the objectives of this study were as follows: (a) To identify and validate appropriate elements and sub-elements of a teaching and learning model that can help improve computational thinking skills with the use of the Delphi technique and (b) To explore the contribution of each element and sub-element to the development of computational thinking skills. The outcomes of the validation process helped lay the foundation for the formulation of the proposed teaching and learning model that would help students to improve their computational thinking skills.

## 2. Materials and Methods

The Delphi technique was used in this study used to address the research objective given its high effectiveness in gathering and analyzing experts' opinions of the field of this study. A survey was carried out for three (3) rounds to ensure the experts would finally reach a strong consensus or agreement of the elements and sub-elements of the proposed teaching and learning model.

### 2.1. Participants

An expert is an individual who has vast experience and strong knowledge of a particular topic. In this study, the selection of experts was based on their expertise and experience in the field of computational thinking. As such, seven (7) experts with 11 years of working experience or more in computer programming and computational thinking-related fields were recruited. Such a number of experts was in line with the recommendations made by [19] and [20], who assert that there will be no significant differences in the consensus reached by a panel consisting of three, five, nine, and 11 experts. Purposeful sampling technique was used to select the experts based on their experiences and involvements in teaching computer programming courses and as master trainers in the field of computational thinking, which resulted in the selection of five (5) academicians and two (2) industrial practitioners.

### 2.2. Assessment Instrument

An assessment instrument was used in this study, the design of which was first carried out by surveying several sources of information regarding elements and sub-elements that were deemed relevant to the development of the proposed learning and teaching model. Essentially, the contents of such sources, such as journal articles, technical reports, Master's theses, doctoral theses, and the websites of certain organizations, were screened

to identify elements relevant to the proposed model.

Five (5) participants with vast experiences in computational thinking and computer programming were selected for a pilot test to validate the research instrument, the results of which helped to improve its structure and contents. Also, the reliability of the assessment instrument was further enhanced by incorporating their feedback in the instrument accordingly. Each item was rated along 5-point Likert-type scales to measure respondents' levels of agreement as summarized in Table 1. The results of the pilot test were also used to guide the development of the assessment instrument which would be used for the implementation of the Delphi technique.

**Table 1.** Assessment scales of experts' opinions

Scale	Description
1	Extremely not imperative
2	Not imperative
3	Moderately imperative
4	Imperative
5	Extremely imperative

**2.3. Implementation Procedure of the Delphi Technique**

In this study, the Delphi technique was carried out for three (3) rounds as detailed in the following sub-sections:

**2.3.1. Round 1**

In the first round, the assessment instrument, named Round1 Assessment Instrument (R1-AI), consisting of seven (7) elements and 17 sub-elements were sent to the experts through official postal mails and emails to elicit their opinions. Their constructive feedback helped to improve the reliability of the instrument by enabling the

researchers to re-examine the instrument and make necessary changes deemed important. The revised instrument would then be subjected to the second round of the Delphi technique.

**2.3.2. Round 2**

The aim of the second round of this technique is to elicit a consensus of experts' opinions. For this purpose, the researchers distributed the revised assessment instrument, named Round2 Assessment Instrument (R2-AI), to the same experts to seek their comments or feedback. The acceptance and rejection of elements and sub-elements was based on a cut-off point of 3.5 of the assessment continuum, ranging from '1' (*Extremely not imperative*) to '5' (*Extremely imperative*). Thus, elements and sub-elements attaining a rating of 3.5 and above would be accepted. Conversely, those with ratings lower than 3.5 would be rejected [21]. The second round of this technique was deemed complete after all elements and sub-elements had been either accepted or rejected. The R2-AI instrument was checked and refined prior to the implementation of the third round of this technique.

**2.3.3. Round 3**

The revised assessment instrument from the second round was renamed Round3 Assessment Instrument (R3-AI). The instrument was distributed to the same experts to elicit their final agreement on the validity of the elements and sub-elements of the revised version. In the third round, a consensus of experts' opinions of the validity of the revised elements and sub-elements of the R3-AI instrument was obtained. Figure 1 shows the implementation of the Delphi technique carried out in this study.

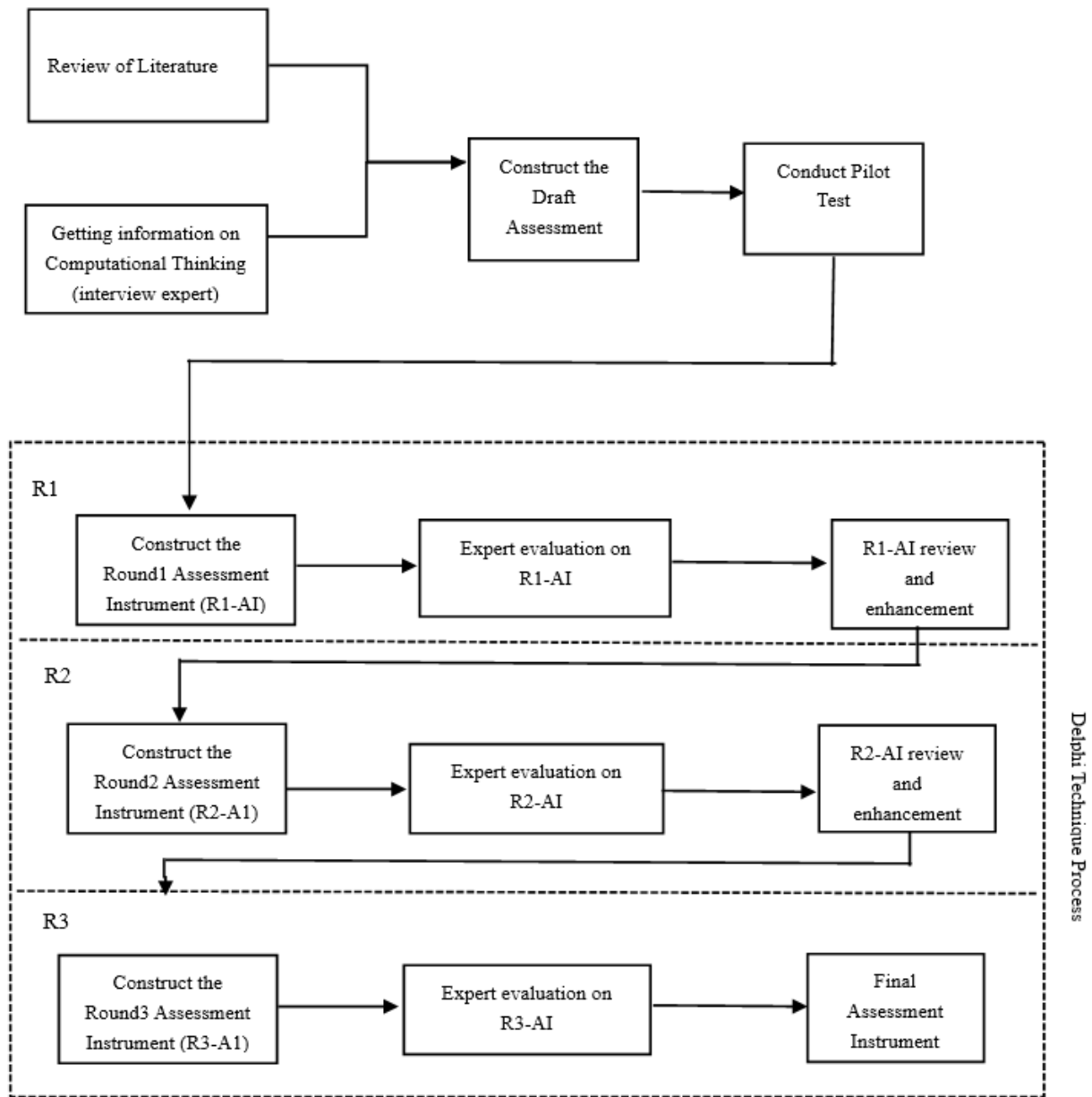


Figure 1. The implementation of the Delphi technique [18] in this study

### 3. Findings and Discussion

The panel of experts recruited in this study was made up of five (5) lecturers who were involved in the teaching of computer programming and were also computational thinking master trainers and two (2) industrial practitioners who were also computational thinking master trainers. All of the experts were vastly experienced whose working experiences ranged from 15 to 25 years. The findings of the first and second rounds of the Delphi technique showed that there were several elements and sub-elements that had to be refined and discarded and new elements and sub-elements had also to be introduced. The results of the third round of this technique helped to establish the validity and reliability of elements and sub-elements. Table 2 summarizes the demographic

backgrounds of the experts.

#### 3.1. Findings of the First and Second Round of the Delphi Technique

Initially, prior to the first two rounds of the Delphi technique, the assessment instrument consisted of seven (7) elements, namely Understanding and Defining Problems (E1), Planning Problem-Solving (E2), Designing Problem-Solving (E3), Coding (E4), Testing (E5), Documenting (E6), and Metacognitive Skills (E7), all of which were made up of 17 sub-elements. After the implementation of these rounds, several changes were made, such as maintaining, refining, adding, and exchanging some of the sub-elements and elements, which are summarized in Table 3. The details of these changes

are as follows:

- (a) Initially, the first element, E1, had three (3) sub-elements as follows: (a) *Defining a problem by reading and understanding it* (E1.1), (b) *Analyzing the problem by identifying the input (the given data), and process (the appropriate problem-solving method), and output (the intended outcome)* (E1.2), and (c) *Identifying the most suitable, efficient problem-solving concept* (E1.3). After the implementation of the first two rounds of the Delphi technique, the first two sub-elements were retained, while the last was re-defined as a sub-element of the second element, E2.
- (b) Originally, the second element, E2, comprised four (4) sub-elements as follows: (a) *Planning several methods of solving a problem and determining whether there are links between them* (E2.1), (b) *Identifying the most suitable, efficient problem-solving concepts* (E2.2), (c) *Choosing the most suitable, efficient problem-solving concept* (E2.3), and (d) *Dividing the problem into sub-problems based on specific functions* (E2.4). Several changes were made after the implementation of the first two rounds of the Delphi technique. For example, the wording of E2.1 was refined to make it more intelligible, while E2.2 was a new sub-element transferred from the first element, E1. The last two sub-elements, namely E2.3 and E2.4, were retained. Another new sub-element was introduced, namely E2.5 (*Adapting or using the original codes to solve a new problem*), which was originally a sub-element of the sixth element, E6.
- (c) Initially, the third element, E3, constituted two (2) sub-elements, namely *Writing appropriate algorithms based on the analysis and planning carried out earlier* (E3.1) and *Examining the written algorithms to ensure accurate output* (E3.2). After the first two rounds of the Delphi technique, no changes were made to these two sub-elements. However, a new sub-element was added to the third element, E3. As such, the third element, E3, now comprises three sub-elements.
- (d) Originally, the title of the fourth element, E4, was 'Authoring Codes', but it was changed to 'Writing Codes', which consists of five (5) sub-elements as follows: (a) *Preparing codes based on algorithms written earlier* (E4.1), (b) *Checking any errors in the program* (E4.2), (c) *Ensuring the program is free from run-time errors* (E4.3), (d) *Ensuring the program is free from logical errors* (E4.4), and (e) *Correcting syntax errors or logical errors*. Subsequently, E4.1 was reworded to 'developing codes' from 'preparing codes', and the wording of E4.5 was appended with the phrase 'if any'. For the remaining sub-elements, no changes were made.
- (e) Earlier, the fifth element, E5, consisted of two sub-elements, namely *Executing the program repetitively by entering different sets of inputs to generate precise output* (E5.1) and *Examining the output in terms of efficiency, reliability, and readability* (E5.2). After the first two rounds of the Delphi technique, E5.2 was split into two sub-elements, namely E5.2 (*Examining the output in terms of readability*)' and E5.3 (*Examining the output in terms of efficiency*). By contrast, no changes were made to E5.1.
- (f) In the initial stage, the sixth element, E6, comprised two (2) sub-elements, namely *Documenting programs, problem-solving strategies, and examples of input and output produced* (E6.1) and *Adapting or using original codes to solve new problems* (E6.2). Subsequently, the former was discarded and the latter transferred to the second element, E2. By taking the experts' opinions into account, this sixth element, E6, would not be considered in the process of formulating the model.
- (g) Originally, the seventh element, E7, constituted only one (1) sub-element, namely *Involving activities such as planning, monitoring, checking, selecting, assessing, self-examining, and self-monitoring* (E7.1). Later, this final element was refined by dividing it into seven (7) sub-elements as follows: (a) *Performing planning activities* (E7.1), (b) *Performing monitoring activities (relating to the monitoring activities in the design and coding phases)* (E7.2), (c) *Performing selecting activities* (E7.3), (d) *Performing checking activities (relating to testing activities)* (E7.4), (e) *Performing assessment activities* (E7.5), (f) *Performing self-monitoring activities* (E7.6), and (g) *Performing self-examining activities* (E7.7).

**Table 2.** The demographic background of the experts

Panel	Occupation	Working experience (in year)	Sector
1	Lecturer	16	Public University
2	Lecturer	24	Public University
3	Lecturer	25	Public University
4	Lecturer	15	Public University
5	Lecturer	18	Public University
6	Industrial practitioner	15	Industry
7	Industrial practitioner	17	Industry

Table 3 summarizes the elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills after the first and second round of the Delphi technique.

In addition to refining, adding, and discarding several elements of the assessment instrument, some experts also suggested that the definition of metacognition had to be made more detail at the beginning of the instrument to

foster a better understanding of such a concept. All the comments and suggestions made by the experts in the first and second round of the Delphi technique were taken into

account by making the necessary changes to the assessment instrument, which would be subjected to the third or final round of the same technique.

**Table 3.** Elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills after the first and second round of the Delphi technique

Elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills		Retained, transferred, and discarded elements	New and refined elements
<b>E1 Understanding and Defining Problems</b>			
E1.1	Defining problems by reading and understanding them.	Retained	
E1.2	Analyzing the problems by identifying the input (the given data), process (the appropriate problem-solving method), and output (the intended outcome).	Retained	
E1.3	Identifying the most suitable, efficient Problem - solving concept.	Transferred to E2.2	
<b>E2 Planning Problem-Solving</b>			
E2.1	Using existing knowledge to plan several problem - solving methods to solve problems as defined by E1.		Refined
E2.2	Identifying the most suitable, efficient problem-solving concept.	Transferred from E1.3	
E2.3	Choosing the most suitable, efficient problem-solving concept.	Retained	
E2.4	Dividing the problem into sub-problems and specific functions.	Retained	
E2.5	Adapting or using the original codes to solve new problems.	Transferred from E6.2	
<b>E3 Designing Problem-Solving</b>			
E3.1	Writing appropriate algorithms based on the analysis and planning carried out earlier.	Retained	
E3.2	Examining the written algorithms to ensure accurate output.	Retained	
E3.3	Checking the developed algorithms to ensure efficient problem-solving.		Refined
<b>E4 Writing Codes</b>			
E4.1	Developing codes based on algorithms written earlier		Refined
E4.2	Checking any syntax errors in the program.	Retained	
E4.3	Ensuring the program is free from run-time errors	Retained	
E4.4	Ensuring the program is free from logical errors.	Retained	
E4.5	Correcting syntax errors or logical errors, if any.		Refined
<b>E5 Testing</b>			
E5.1	Executing the program repetitively by entering different sets of inputs to generate precise output.	Retained	
E5.2	Examining the output in terms of readability.		Refined
E5.3	Examining the output in terms of efficiency.		Refined
<b>E6 Documenting</b>			
E6.1	Documenting programs, problem-solving strategies, and examples of input and output produced.	Discarded	
E6.2	Adapting or using the original codes to solve new problems.	Transferred to E2.5	
<b>E7 Metacognitive Skill</b>			
E7.1	Performing planning activities.		Refined
E7.2	Performing monitoring activities (relating to the monitoring activities in the design and coding phases)		Refined
E7.3	Performing selecting activities.		Refined
E7.4	Performing checking activities (relating to testing)		Refined
E7.5	Performing assessing activities.		Refined
E7.6	Performing self-monitoring activities.		Refined
E7.7	Performing self-examining activities.		Refined

### 3.2. Findings of the Third Round of the Delphi Technique

Table 4 shows the revised elements and sub-elements of the teaching and learning model for the improvement of computational thinking skill after the third round of the Delphi technique. As shown, there were six (6) elements and 25 sub-elements that had been validated which would be embedded in the proposed model as follows: (a) Understanding and Defining Problems (E1), (b) Planning Problem-Solving (E2), (c) Designing Problem-Solving (E3), (d) Writing Codes (E4), (e) Testing (E5), and (f) Metacognitive Skill (E6). Table 4 also shows the mean scores and standard deviations of the elements and sub-elements of the proposed model.

The followings are the detailed description of the embedded elements and sub-elements of the teaching and learning model for the improvement of computational thinking skill.

- (a) As shown, the mean score of the second element, E2, was 5, which was extremely high, signifying that it is an extremely important element that is needed to improve students' computational thinking skills. Specifically, the first sub-element, E2.1 (*Using existing knowledge to plan several problem-solving methods to solve problems*), and the fifth sub-element, E2.5 (*Adapting or using the original codes to solve new problems*) are highly important because they are consistent with the concept of generalization of computational thinking, which is a concept of solving a new problem based on previously-solved problems. Likewise, the fourth sub-element, E2.4, is also highly important as it divides a problem into several manageable sub-problems and functions, which closely relates to the concept of decomposition of computational thinking that breaks down a complex problem into smaller problems that are easily discerned and managed [22]. Thus, the implementation of the planning activities of the second element (E2) of the proposed model can help develop the two sub-components of computational thinking skill, namely generalization and decomposition.
- (b) Similarly, the findings suggest that the third element, E3 (*Designing problem-solving*), is highly important too, entailing it to be embedded in the teaching and learning model for the improvement of computational thinking skill. Essentially, this element is closely related to the algorithmic element of computational thinking where an algorithm refers to a series of structured commands to solve a problem or to perform logical reasoning [23]. Specifically, the second sub-element, E3.2 (*Examining the written algorithms to ensure accurate output*), and the third sub-element, E3.3 (*Checking the developed algorithms to ensure efficient problem-solving*), are in line with the concept of assessment of computational thinking, which is to ensure the solution of an algorithm, system or process is precise and suitable for its intended use [22]. In particular, the activities carried out during the designing process to generate computer programs relate to algorithmic thinking of computational thinking. Thus, as students perform designing activities to produce computer programs, they are also developing their computational thinking skills. This finding is consistent with those of other studies, clearly emphasizing the importance of this element in improving the algorithmic thinking and assessment of students' computational thinking [23]. As such, this element should be embedded in the proposed teaching and learning model, given the high rating of its sub-elements, with all attaining a mean score of 5.
- (c) The findings showed that the fourth element, E4, was also deemed highly important and valid by the experts, with four of its sub-elements attaining a mean score of 5, while the remaining sub-element achieving a mean score of 4.86. Such findings are hardly surprising given that one of its sub-element is closely related to the concept of decomposition, algorithmic thinking, and assessment of computational thinking. In this regard, coding activities entail strong logical thinking and high creativity among students to generate precise, quality computer output. Thus, the implementation of the fourth element, E4, in the teaching and learning process can help improve students' computational thinking in terms of decomposition, algorithmic thinking, and assessment. In light of this revelation, element E4 together with its sub-elements should be embedded in the proposed teaching and learning model to ensure its effectiveness in developing students' computational thinking skills.
- (d) The findings also indicated that sixth element, E6 (consisting of seven sub-elements), was highly rated by the experts, with five of its sub-elements attaining a mean score of 5. Again, this specific finding is not surprising as all the sub-elements of E6, namely E6.1, E6.2, E6.3, E6.4, and E6.5 (which involve planning, monitoring, selecting, checking, and assessing activities) are closely related to the specific elements of computational thinking, namely decomposition, algorithm, generalization, and evaluation. Thus, such an element is crucial to the development of students' computational thinking skills. Likewise, the same sub-elements have been studied by other researchers that focused on computer programming [12-16], indicating its significance in problem-solving techniques. On the other hand, the remaining sub-elements, namely E6.6 and E6.7, attained mean scores of 4.86 and 4.71, respectively, underscoring

their importance in the development of students' socialization skills. Overall, the above findings suggest that element E6 is one of the important elements that needs to be embedded in the proposed teaching and learning model.

- (e) As shown in Table 4, the first element, E1 (consisting of two sub-elements), was also deemed reliable and valid by the experts, as evidenced by the high ratings of its sub-elements, namely E1.1 (*Defining a problem by reading and understanding it*) and E1.2 (*Analyzing the problem by identifying the input, process, and output*), with the former and latter attaining mean scores of 4.7 and 5.0, respectively. Arguably, such a finding is tenable as this element is closely related to one of the important aspects of computational thinking, namely

abstraction, an ability that helps the abstraction of data that are only relevant to a problem and eliminates those deemed irrelevant. Surely, the inclusion of this element in the proposed teaching and learning model can help students develop such an important skill or ability. Such an inclusion can be justified by the evidence provided by the findings of previous studies, in which several researchers used this element in the teaching of computer-related courses which helped students to develop strong abstraction ability or skill [12, 23-25].

Table 4 summarizes the elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills after the final round of the Delphi technique.

**Table 4.** Elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills after the final round of the Delphi technique

Elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills		Experts' assessment								Mean	Std. Dev.	Status	
		L1	L2	L3	L4	L5	L6	L7					
<b>E1 Understanding and Defining Problems</b>													
E1.1	Defining problems by reading and understanding them [7-11].	4	4	5	5	5	5	4			4.57	0.463	Accepted
E1.2	Analyzing the problems by identifying the input (the given data), process (the appropriate problem-solving method), and output (the intended outcome) [8-9, 11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
<b>E2 Planning Problem-Solving</b>													
E2.1	Using existing knowledge to plan several problem-solving methods to solve problems as defined by E1 [7 - 9, 11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
E2.2	Identifying the most suitable, efficient problem-solving concept [7-9, 11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
E2.3	Choosing the most suitable, efficient problem-solving concept [7-9, 11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
E2.4	Dividing the problem into sub-problems and specific functions [9-11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
E2.5	Adapting or using the original codes to solve new problems [7-11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
<b>E3 Designing Problem-Solving</b>													
E3.1	Writing appropriate algorithms based on the analysis and planning carried out earlier [7-9].	5	5	5	5	5	5	5			5.00	0.000	Accepted
E3.2	Examining the written algorithms to ensure accurate output [7-11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
E3.3	Checking the developed algorithms to ensure efficient problem-solving [7-11].	5	5	5	5	5	5	5			5.00	0.000	Accepted
<b>E4 Writing Codes</b>													



Table 4 Continued

E4.1	Developing codes based on algorithms written earlier [7-9].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E4.2	Checking any syntax errors in the program [7-9, 11].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E4.3	Ensuring the program is free from run-time errors [7-9, 11].	4	5	5	5	5	5	5		4.86	0.327	Accepted
E4.4	Ensuring the program is free from logical errors [7-9, 11].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E4.5	Correcting syntax errors or logical errors, if any [7-9, 11].	5	5	5	5	5	5	5		5.00	0.000	Accepted
<b>E5 Testing</b>												
E5.1	Executing the program repetitively by entering different sets of inputs to generate a precise output [7-11].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E5.2	Examining the output in terms of readability [7-11].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E5.3	Examining the output in terms of efficiency [7-11].	4	4	4	4	4	4	4		4.00	0.000	Accepted
<b>E6 Metacognitive Skill</b>												
E6.1	Performing planning activities [12-16].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E6.2	Performing monitoring activities (relating to the monitoring activities in the design and coding phases) [12-16].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E6.3	Performing selecting activities [12-16].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E6.4	Performing checking activities (relating to testing) [12-16].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E6.5	Performing assessing activities [12-16].	5	5	5	5	5	5	5		5.00	0.000	Accepted
E6.6	Performing self-monitoring activities [12-16].	5	5	5	5	5	4	5		4.86	0.327	Accepted
E6.7	Performing self-examining activities [12-16].	5	5	4	5	5	5	4		4.71	0.423	Accepted
	Overall									4.79	0.410	

(f) The same findings also indicated that the fifth element, E5 (consisting of three sub-elements), received a strong consensus regarding its validity from the experts, with its first two sub-elements, E5.1 (*Executing the program repetitively by entering different sets of inputs to generate a precise output*) and E5.2 (*Examining the output in terms of readability*), attaining a mean score of 5 each. By contrast, with a mean score of 4, the last sub-element of this element, E5.3 (*Examining the output in terms of efficiency*), was deemed moderately important. These findings suggest that all the sub-elements, especially the first two, are important to help improve students' computational thinking skills, entailing their inclusion in the proposed teaching and learning model. However, the experts cautioned that the third sub-element E5.3, must not be overly emphasized for novice students, as they were quite new to the concept of computational thinking. As such, only basic aspects of computational thinking

should be exposed to such students.

Overall, these findings highlight the importance of this testing element, E5, which is closely related to two important components of computational thinking, namely evaluation and generalization. To lend credence to such revelations, similar findings had been reported in previous studies [26-29], which showed the element of testing was important to help improve students' computational thinking skills.

As a whole, the findings discussed in this section helped the researchers to address the research objectives, which were to validate the elements and sub-elements of a teaching and learning model for the improvement of computational thinking skill with the use of the Delphi technique and to examine the relations between the proposed model' elements and sub-elements and those of computational thinking skill. Table 5 highlights the mapping of the elements and sub-elements of the proposed model and computational thinking skill.

**Table 5.** The mapping of the elements and sub-elements between the proposed model and computational thinking

Elements and sub-elements of the teaching and learning model for the improvement of computational thinking skills		Computational Thinking' Elements				
		Abstraction	Decomposition	Algorithm	Evaluation	Generalization
<b>E1 Understanding and Defining Problems</b>						
E1.1	Defining problems by reading and understanding them.	✓				
E1.2	Analyzing the problem by identifying the input (the given data), process (the appropriate problem-solving method), and output (the intended outcome).	✓				
<b>E2 Planning Problem-Solving</b>						
E2.1	Using existing knowledge to plan several problem-solving methods to solve problems as defined by E1.					✓
E2.2	Identifying the most suitable, efficient problem-solving concept		✓			
E2.3	Choosing the most suitable, efficient problem-solving concept.					✓
E2.4	Dividing the problem into sub-problems and specific functions.		✓			
E2.5	Adapting or using the original codes to solve new problems.					✓
<b>E3 Designing Problem-Solving</b>						
E3.1	Writing appropriate algorithms based on the analysis and planning carried out earlier.			✓		
E3.2	Examining the written algorithms to ensure accurate output.			✓	✓	
E3.3	Checking the developed algorithms to ensure efficient problem-solving.			✓	✓	
<b>E4 Writing Codes</b>						
E4.1	Developing codes based on algorithms written earlier.			✓		
E4.2	Checking any syntax errors in the program.				✓	
E4.3	Ensuring the program is free from run-time errors.				✓	
E4.4	Ensuring the program is free from logical errors.				✓	
E4.5	Correcting syntax errors or logical errors, if any.		✓			
<b>E5 Testing</b>						
E5.1	Executing the program repetitively by entering different sets of inputs to generate a precise output.					✓
E5.2	Examining the output in terms of readability.				✓	
E5.3	Examining the output in terms of efficiency.				✓	
<b>E6 E6: Metacognitive Skill</b>						
E6.1	Performing planning activities.		✓			✓
E6.2	Performing monitoring activities (relating to the monitoring activities in the design and coding phases).		✓	✓	✓	
E6.3	Performing selecting activities.			✓		
E6.4	Performing checking activities (relating to testing).					✓
E6.5	Performing assessing activities				✓	
E6.6	Performing self-monitoring activities.	E6.6 and E6.7 are two new elements added to the teaching and learning model				
E6.7	Performing self-examining activities.					

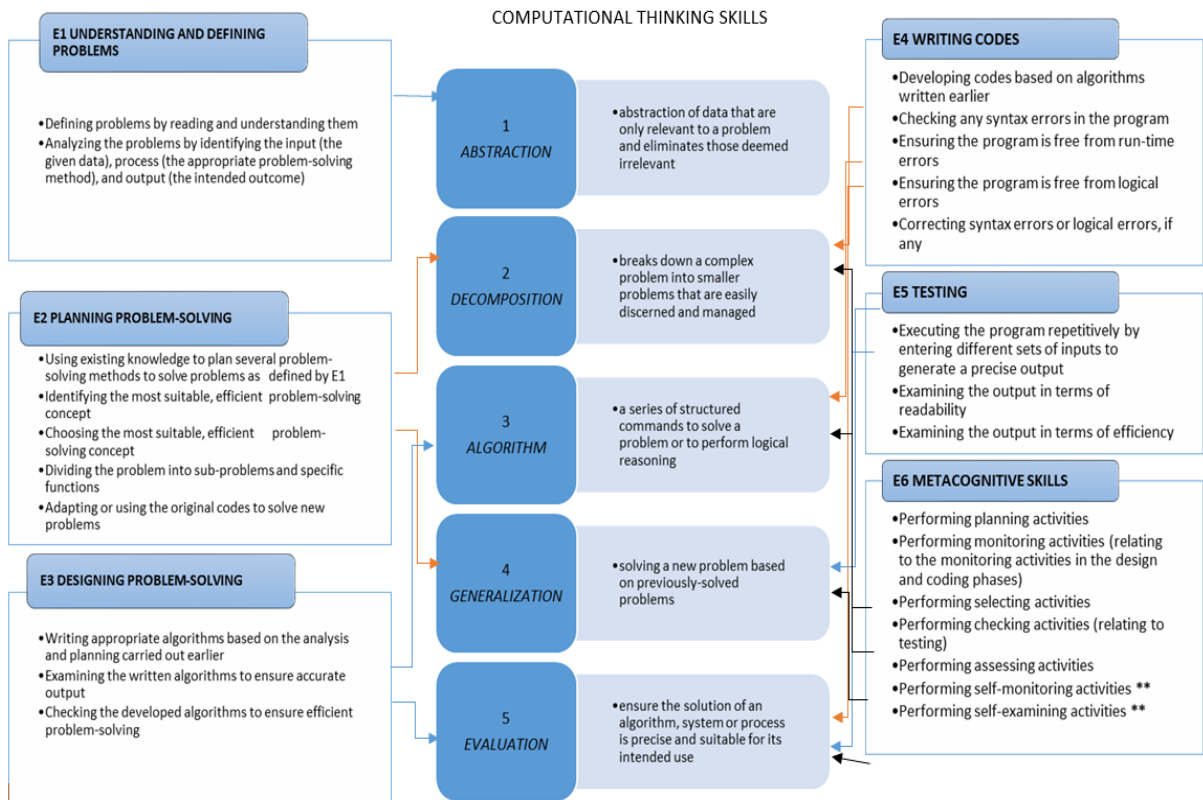
The summary of the findings of the Delphi technique used to validate the elements and sub-elements of the proposed teaching and learning model are as follows:

- (a) Prior to implementing the Delphi technique, the assessment instrument consisted of seven (7) elements and 17s sub-elements. Subsequently, after applying such a technique, several changes were made to the proposed model, which by then had six (6) elements and 25 sub-elements.
- (b) The most important elements were E2 and E3, followed by E4, E6, E1, and E5.
- (c) The above elements and sub-elements, which had been vigorously validated by a panel of experts, should be embedded in the proposed teaching and learning model, which could be used to guide the development of learning activities to improve students’ computational thinking skills.

in twenty-first century learning. Thus, it needs to be cultivated as early as possible, but teaching practitioners have been struggling to find effective techniques to be used in the teaching and learning process that can help students to develop such a skill. Against this backdrop, this study was carried out to develop and validate the elements and sub-elements of a proposed teaching and learning model for such a purpose. Specifically, such elements and sub-elements were conceptualized from the combination of elements of problem-solving and metacognition, which are two popular techniques used in the teaching and learning of computer programming. After a rigorous validation process through the Delphi technique, six (6) elements and 25 sub-elements were found to be highly valid. In conclusion, the findings of this research can be used to guide the development of an effective teaching and learning model with which teaching practitioners can use to select the most appropriate technique for their teaching practices. This study contributes to the development of the teaching and learning model for enhancing students’ computational thinking skills as illustrated in Figure 2.

### 4. Conclusions

Every student should acquire strong computational thinking skill, which is a new but important skill needed



Note: \*\* related to enhance the social and thinking skill

**Figure 2.** Teaching and Learning Model for enhancing students’ Computational Thinking Skills thru Computer Programming

## Acknowledgments

The authors wish to extend their gratitude to the Ministry of Education of Malaysia for the Fundamental Research Grant [FRGS/1/2018/SS109/UPSI/02/29] that helped to fund this research and to the Research Management and Innovation Centre of UPSI for the support rendered.

## REFERENCES

- [1] Li, Y. (2016, October). Teaching programming based on Computational Thinking. In *Frontiers in Education Conference (FIE), 2016 IEEE* (pp. 1-7). IEEE.
- [2] M. Román-González, J.C. Pérez-González, C. Jiménez-Fernández, Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, vol. 72, pp. 678-691, 2017.
- [3] C. C. Selby. Relationships: computational thinking, pedagogy of programming, and bloom's taxonomy, in *Proceedings of the Workshop in Primary and Secondary Computing Education on ZZZ*, pp. 80–87, 2015.
- [4] N. Jalal, Z. Lechen, M. Linda, N. Eva. Development of computational thinking, digital competence and 21st century skills when learning programming in K-9, *Education Inquiry*, vol. 11, no. 1, pp. 1-17, 2020, DOI: 10.1080/20004508.2019.1627844
- [5] F. Quitério, A. José. How to Improve Computational Thinking: a Case Study Education in the Knowledge Society, *Espana*, vol. 18, no. 4, pp. 35-51, 2017.
- [6] M. Bower, L. N. Wood, J. W. Lai, C. Howe, R. Lister, R. Mason, K. Highfield, J. Veal. Improving the Computational Thinking Pedagogical Capabilities of School Teachers. *Australian Journal of Teacher Education*, vol 42, no. 3, 2017. <http://dx.doi.org/10.14221/ajte.2017v42n3.4>
- [7] A. E. Kursat , K. Hasan, S. Guven , A. A. Dilara, A. G. Aysegul, I. B. Ali. A Model Suggested for Programming Teaching: Programming in Seven Steps, *Education and Science*, vol 44, no. 197, pp. 155-183, 2019.
- [8] S. I. Malik, J. A. Coldwell-Neilson. A model for teaching an introductory programming course using ADRI, *Educ Inf Technol*, vol. 22, pp. 1089-1120, 2017. <https://doi.org/10.1007/s10639-016-9474>.
- [9] P. D. Fadi, T. Murray, A. M. James. A Common Model for Problem Solving and Program Development, *IEEE Transactions on Education*, vol. 42, no. 4, pp. 331-336, 1999.
- [10] G. Polya. *How to solve it: A New Aspect of Mathematical Method*, (2<sup>nd</sup> edition), Doubleday & Company Inc, Garden City, New York, USA. 1945.
- [11] D. P. Fadi. An integrated environment for problem solving and program development, PhD Dissertation, 1033, New Jersey Institute of Technology, 1997.
- [12] M. Havenga. The role of metacognitive skills in solving object-oriented programming problems: a case study, *TD the Journal for Transdisciplinary Research in Southern Africa*, vol. 11, no. 1, pp. 133-147, 2015.
- [13] M. R. Siti Nurulain. A Metacognitive Support Environment for Novice Programmer Using Semantic Web, PhD Thesis, Universiti of Malaya, Kuala Lumpur, 2015.
- [14] A.H. Abdullah, S.N.S.A. Rahman, M.H. Hamzah. Metacognitive Skills of Malaysian Students in Non-Routine Mathematical Problem Solving. *Bolema, Rio Claro (SP)*, vol. 31, no. 57, pp. 310 – 322, 2017.
- [15] M. Gaeta, G. R. Mangione, F. Orciuoli, S. Saverio. Metacognitive Learning Environment: a semantic perspective, *Journal of e-Learning and Knowledge Society*, vol.7, no. 2, pp. 69-80, 2014.
- [16] P.M. Nimmi, K.A. Zakkariya. Developing Metacognitive Skills: A Potential Intervention for Employability Enhancement, *Journal of Contemporary Research in Management*, vol. 11, no. 3, pp. 11-20, 2016.
- [17] E. K. Thomas. Approaches for Developing a Model for Information and Communication Technology (computational thinking) Implementation in the Higher Education Environment, PhD Dissertation, College of Engineering and Computing, Nova Southeastern University, USA, 2015.
- [18] J. Nworie. Using the Delphi technique in educational technology research, *Tech Trends - September*, vol. 55, no. 5, pp. 22-30, 2011.
- [19] K. Brockhoff. The performance of forecasting groups in computer dialogue and face- to-face discussion. In Linstone, H. & Turoff, M. (Eds.), *The Delphi methods: technique and applications*. London: Addison-Wesley, 1975.
- [20] G. Rowe & G. Wright. Expert opinions in forecasting: the role of the Delphi technique. *Principles of Forecasting*, 125-144, 1999.
- [21] L. T. Hai, P. H. Hai, C. L. Thai, J. Huge, A. Ahenkan, L. X. Quynh, & V. Van Hieu. Software for sustainability assessment: a case study in Quang Tri province, Vietnam. *Environmental Modeling & Assessment*, vol 16, no. 6, pp. 541–550, 2011.
- [22] S. Bocconi, A. Chiocciariello, G. Dettori, A. Ferrari, K. Engelhardt. Developing computational thinking in compulsory education – Implications for policy and practice, 2016, doi:10.2791/792158.
- [23] U. Nor Hasbiah, H. Jamilah. A Web-based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?, *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9, 1945-1958, 2019.
- [24] B. Xabier, A. O. Miguel, C. O. Juan, J. R. Mauricio. Computational Thinking in pre-university Blended Learning Classrooms, *Computers in Human Behavior*, vol.80, 2018.
- [25] R. Margarida, L. Alexandre, L. Benjamin, Computational thinking development through creative programming in higher education, *International Journal of Educational Technology in Higher Education*, vol. 14, no. 42, 2017.
- [26] L. L. Ung, C. S. Tammie, L. Jane, A. A. Norazila. Preliminary Investigation: Teachers' Perception on

Computational Thinking Concepts, *Journal of Telecommunication and Computer Engineering*, vol. 9, pp. 2-9, 2017.

- [27] S. Aslina. Students' Computational Thinking Skill through Cooperative Learning Based on Hands-on, Inquiry-based, and Student-centric Learning Approaches, *Universal Journal of Educational Research*, vol. 8, no. 1, pp. 290-296, 2020.
- [28] T. Djambong, V. Freiman, Task-Based Assessment of Students' Computational Thinking Skills Developed Through Visual Programming or Tangible Coding Environments, in *Proc. of 13th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2016)*, pp. 41-52, 2016.
- [29] D. Hooshyar, R. B. Ahmad, M. Yousefi, F. D. Yusop, S. J. Horng. A flowchart-based intelligent tutoring system for improving problem-solving skills of novice programmers, *Journal of Computer Assisted Learning*, pp. 1-7, 2015.