

A Web-based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

Nor Hasbiah Ubaidullah, Jamilah Hamid

Abstract: *Of late, a number of studies have been carried out to infuse the concept of Computational Thinking into programming courses, given that such a concept and programming are closely intertwined. Admittedly, most instructors find teaching programming to be very challenging, especially to novice students. Currently, the dominant teaching approaches used to teach computer programming mainly involve lectures, discussions, and lab practical sessions. Nevertheless, the use of such approaches has been heavily criticized for their ineffectiveness in developing skillful students because of the minimal interaction between students and teachers, especially in large classes. Coupled with the lack of proper teaching aids, the problem is further aggravated, putting the students in a dire situation. Against this backdrop, the authors carried out a study to examine the prevailing problems in the current teaching technique used by programming instructors in teaching programming to novice students, to determine programming instructors' knowledge about the concept of CT, and to seek a new appropriate teaching aid for teaching programming. A qualitative research method was used involving a semi-structured interview, in which 10 instructors from ten pre-university centers in Malaysia were recruited as the interviewees. The research findings showed that existing teaching approaches used were not effective as students had to rely on additional time to learn programming. In addition, it was found that a majority of the interviewee were fully acquainted with the concept of CT. Moreover, the interviewees professed a strong need for a portal which could be used anywhere, anytime to help novice students to learn programming and enhance their CT skill. Together, these two important findings underscore the imperative to introduce a new teaching approach, preferably aided by a novel application, which can be used by novice students to enhance their programming and CT skills. In this paper, the authors propose an instructional design model for the development of a learning application to help such students.*

Index Terms: *Computational thinking, novice student, programming, portal.*

I. INTRODUCTION

In computer science and Information Technology program, computer programming is one of the core, mandatory courses that students need to learn and master. However, learning such a course has been found to be daunting to many students, with many lecturers citing that the former had poor problem-solving skills in using the proper syntaxes and

semantics of the programming language, the impact of which made the students incapable to solve programming problems. Put simply, the students seemed lacking the Computational Thinking (CT) skill [1] - [3]. In this regard, according to Park [4], CT is a creative skill in solving programming problems involving computer hardware and software. In a more detailed account, Wing [5] argues that, in actuality, individuals use CT when they use logical thinking to solve intricate, complex problems by deconstructing them into smaller units to be solved mathematically, which ultimately leads to the development of computer applications. In particular, he asserts that CT is something that is scientific that helps create command-line algorithms with which a computer can execute specific tasks in solving problems. As such, programming serves as a medium by which CT is applied through the application of systems and equipment in various disciplines [6]. To address the diverse definitions of such a construct, Selby and Woolaard [7] argue that CT refers to "a definition based on elements that are widely accepted: abstraction, decomposition, evaluations, generalization, and algorithmic thinking". Likewise, a UK-based organization (Barefootcas.org.uk) offers an interesting explanation of such a concept, stating that "to this end, there are various CT concepts that could be implemented in the classroom, such as logic, algorithm, decomposition, pattern, abstraction, and evaluation" [8], a notion shared by MDEC [9]. On a broader perspective, CT is closely linked to science, technology, and society because it involves logical, analytical, and systematic processes in solving problems faced by humans [10]. Given such definitions, it is clear that CT is important in people's lives, which needs to be developed and nurtured. However, despite the recognition of its importance, studies that focus on improving students' CT have been seriously lacking, which have been highlighted by Voogt et al., [6] and Wilson and Guzdial [11] who argue that "research on the integration of computational thinking in education is still limited". As such, more efforts have to be put in place to mitigate this predicament. Of late, several researchers have begun conducting studies to focus on such a skill. For example, studies by Margarida, Alexandre and Benjamin [12], Cagin, Mary, Liz, and Lachlan [13], and Xabier, Miguel, Juan, and Mauricio [14] have been carried out to infuse the concept of CT into a programming course, given that such a concept and programming are closely intertwined. In particular, Margarida et al. [12] provide a strong contention that "at different educational levels, CT skill can be developed and assessed through ill-defined problems".

Revised Manuscript Received on July 06, 2019.

Nor Hasbiah Ubaidullah, Computing Department, Universiti Pendidikan Sultan Idris, Tg. Malim Perak, Malaysia (corresponding author).

Jamilah Hamid, Computing Department, Universiti Pendidikan Sultan Idris, Tg. Malim Perak, Malaysia.



A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

Arguably, a number of skills can help develop CT, such as the ability to formulate problems, make decisions, and solve problems creatively [15]. Viewed from the computational perspective, the ability of humans to solve problems using logical thinking and skills in writing codes helps contribute to the creation of computer applications and equipment, which, in essence, reflects the successful use of CT [16]. Essentially, programming novice students can be defined those students who lack knowledge and skills in programming. It is, therefore, hardly surprising to note that such students have to struggle to learn the subject matter, which is primarily attributed to their

(i) limited ability to grasp the knowledge in abstract terms. Inevitably, they tend to face mounting problems in applying the programming concepts that they have learned to develop computer programs [17-19]. In fact, they usually will require more time to fully comprehend the proper way of implementing programming concepts in a particular programming language. In the computing realm, it is common knowledge that the limited ability of novice students in understanding the abstract nature of the programming knowledge is not helping them to learn how to write programs. Apparently, such a problem lies in the way they learn programming – they learn by merely focusing on the codes written line-by-line, not on seeing the overall picture of the structure of a program. Moreover, the knowledge of novices tends to be context-specific rather than general [20], rendering them incapable to apply such knowledge in unfamiliar, new contexts. This contention seems to concur with Siti Nurulain's assertion [21], stating that “students may be good in expressing their understanding of programming concepts (e.g. the function of a pointer), but still they are not able to apply it appropriately”. As such, teachers and lecturers are compelled to repeat their teaching lessons on certain topics several times to ensure students would finally understand the learning contents. Meanwhile, teachers and lecturers have no sufficient time to give extra classes as they have to complete the syllabus of such a course as scheduled. As such, the use of appropriate teaching aids, such as courseware, can certainly help students to learn difficult topics at their own time and place. Put simply, such learning aids can help establish self-paced or self-regulated learning to enable to learn based on their ability and pace, making the learning process more efficient and livelier. Definitely, in learning programming, students need to have not only a strong understanding of programming concepts but also sound cognitive skills, including planning, reasoning, problem-solving, and analytical skills [22]. As such, lecturers will find teaching programming to first-year computer science students extremely challenging as the latter will find learning and understanding the semantics of programming to be cognitively daunting [23].

(ii) Problem Solving and Abstract Thinking. On numerous occasions, studies have shown that analytical skills and problem-solving skills are lacking among students in learning computer programming [24] – [27] and [16]. Likewise, they have been found to have limited ability to comprehend some aspects or concepts of programming that are inherently abstract in nature. Principally, problem-solving in computer programming involves cognitive skills that enable students to work in a methodical manner in building mental representations [28], which leads to the creation of an environment that is conducive for developing metacognition.

According to Siti Nurulain [21], “learning programming helps students to critically re-evaluate the solutions to programming problems that they have derived, which essentially involves creative thinking that can assist them to solve new, unfamiliar problems in a different situation”. Moreover, problem-solving skills include making inferences and thinking analytically to help analyze a given problem. In this respect, students devoid of these skills can render them lost or frustrated in pursuing such a course [21]. Clearly, learning programming entails students to have strong cognitive skills to help them understand, interpret, and solve cognitively challenging programming assignments or tasks [29]. Therefore, in addition to having a sound understanding of a particular programming language, students need to have strong problem-solving, logical, and analytical skills. Certainly, to help students develop such skills, lecturers and teachers need to stimulate their students' thinking by repeating their lessons, demonstrations or explanations. In view of this need, appropriate teaching aids can be used to help stimulate students' minds to think analytically and logically. Interestingly, a study by Masyarah [16] has shown that there were several factors confounding the learning of programming among students, such as the lack of understanding of programming concepts, the difficulty in solving problems, and the nature of programming itself that was difficult to comprehend. Arguably, the use of appropriate strategies can help minimize the problems faced by students. As such, it becomes the imperative of lecturers and teachers to strategize their teaching in ways that can help their students to learn programming more effectively. Essentially, with effective strategies, they can help explain complex programming concepts or principles more clearly to their students [30] – [32]. Over the recent years, a number of studies have been carried out that focused on seeking effective methods to help improve students' understanding of programming concepts, which are highly abstract in nature [16], [30] -[32]. Currently, the dominant teaching approaches used to teach computer programming mainly involve lectures, discussions, and lab practical sessions [33]. The use of such approaches has been heavily criticized as they are only good at producing passive students, especially in big student classes, because of the minimal interaction between students and teachers [2]. For effective teaching, Uysal [34] implores that computer programming must be used together with the problem-solving strategy. Likewise, Mason [35] asserts that the lessons of computer programming must employ a systematic instruction of design activities that provide appropriate learning opportunities and feedback. To date, most teaching models or methods only support personal communication; as such, other methods should be explored to improve the teaching of programming. In this respect, Mason [35] asserts that novel technologies, such as e-learning systems, should be used to improve student learning by making lectures or demonstrations more interactive. In recent years, e-learning has become a popular learning method used in many educational organizations, such as universities and colleges, to help facilitate and improve the learning process with the use of a host of computer and communication hardware. In general, e-learning is a generic term that refers any learning methods that are carried out electronically or digital, including internet- and intranet-based learning



(web-based learning), computer-based learning, virtual classrooms, and digital cooperation. In general, learning contents are typically provided in the form of audio and video cassettes, CD-ROMs, and DVDs and through the internet, intranets, and satellites. In fact, e-learning, which is commonly referred to as online learning or web-based learning, has become commonplace nowadays, given the advent of powerful, but affordable, computers [23]. Research has shown that such a learning approach offers a number of benefits, notably in making the learning process more interactive, efficient, and motivational. With such an approach, learning has become more pervasive, offering students access to learning materials transcending temporal and physical barriers. More interestingly, learning becomes more dynamic and collaborative as students and educators can upload learning materials to the e-learning system, with which they can exchange and share with one another. Furthermore, most e-learning systems support real-time monitoring of students' activities, including simulations, audio and video interactions, small-group works, among other [36]. Lately, efforts to improve e-learning systems are gaining more traction, with many studies being devoted to developing such systems that are more secure, stable, and reliable. Nowadays, virtually all students are using online sources to obtain information and materials that are needed for their assignments, projects or tasks. For example, they can refer to Khan's Academy or watch a video of a particular topic at TED Talk website to seek relevant information. Clearly, such online sources can help students learn outside their normal school hours, allowing them to learn at their own pace. Equally important is that online learning enables students to seek additional assistance to help them learn difficult topics by focusing on specific concepts rather than on general ideas [23]. Admittedly, online learning provides students with ample opportunities to access learning materials; however, there is a pressing need to conduct empirical studies that focus on how to best perform quality e-learning involving difficult courses or subject, such as Information Technology [37]. At the international level, several researchers have carried out a number of studies that focused on CT, such as *Computational thinking development through creative programming in higher education* by Margarida, Alexandre, and Benjamin [12], *Computational thinking development through creative programming in higher education* by Cagin, Mary, Liz, Lachlan [13], *A serious game for developing computational thinking and learning introductory computer programming* by Xabier, Miguel, Juan, and Mauricio [14], and *Computational Thinking in pre-university Blended Learning Classrooms*. In the Malaysian context, discourses concerning the concept of CT are gaining enough momentum to attract the attention of various stakeholders. For example, the Ministry of Education of Malaysia has taken a bold step by integrating such a concept in the curriculum of the educational standard, which takes into effect since 2017 [38]. In principle, the aim of such a new curriculum is to introduce students with the basic skills in computer science and to integrate problem-solving skills, logical thinking, and lifelong learning [39]. For example, the concept of CT, such as logical thinking, evaluation, algorithm design, and abstraction, has been infused into the new curriculum involving all subjects from the elementary-school level (Level 1) to the high-school level [40] - [41]. Given the awareness of the importance of the concept and skill of CT at the school level, it is, therefore,

vital that university academics develop the same level of awareness. To date, several Malaysian researchers have carried out a number of studies focusing on CT [42], [36], but theirs were limited to teachers and schools, with almost none being devoted to studying such a concept at the university level. Against such a backdrop, this study was conducted by focusing on the implementation of CT skills through a web-based learning programming portal to help develop such skills among novice programming students at the pre-university level.

From the above discussion, several themes emerged that helped the researchers to formulate four research objectives as follows:

- a) To examine the current teaching technique used by programming instructors in teaching programming among novice students,
- b) To examine the problems associated with the current teaching technique,
- c) To determine programming instructors' knowledge regarding the concept of computational thinking,
- d) To examine programming instructors' perceptions of the development and implementation of a web-based learning programming portal as a tool to enhance computational thinking skill among novice students.

Correspondingly, four research questions were formulated to address the research objectives.

- a) What are the current teaching techniques used by programming instructors in teaching programming to novice students?
- b) What are the problems encountered with the current teaching techniques?
- c) To what extent the programming instructors know about computational thinking concept?
- d) What are perceptions of the programming instructors with regard to the development and implementation of a web-based learning programming portal as a tool to enhance computational thinking skill among novice students?

II. RESEARCH METHOD

Essentially, a research methodology is a systematic way of finding a solution to address the research questions [43]. As such, this study used a qualitative method using a semi-structured interview with the main aim of collecting the relevant data of the research. Such a technique helped the researchers to obtain detailed, comprehensive answers from the research subjects. Specifically, through the interview, relevant information and data were gathered and analyzed, which helped the researchers conceptualize the appropriate model for the development of a learning application.

A. Instrument

The main instrument of this research was based on two sets of semi-structured questions. In addition, the researchers developed a checklist to help guide the interview process such that the interview would proceed smoothly and deliberately, eliciting the essential responses from the interviewees to answer the research questions [38]. Furthermore, such a checklist was designed based on the funnel analogy to guide the development of a questions list, with general questions and specific questions



A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

occupying the top half and the lower half of the list, respectively. A number of scholars and experts conversant in qualitative research were recruited to ascertain the clarity and appropriateness of such an instrument. The first section helped gather information pertaining to the current teaching aids used by the instructors in teaching programming course; the second section dealt with the recommended features of a learning portal deemed important by such instructors. Specifically, the information obtained from the instructors during the interview session was based on their responses, opinions, and ideas.

B. Procedure

The procedure used in this study involved two phases as follows:

Phase 1: In this phase, a formal letter from the director of the selected pre-university center was secured prior to the interview session.

Phase 2: In this phase, the chosen programming instructors of such a center were interviewed and their responses were recorded and then transcribed accordingly. The interview sessions lasted for not more than an hour.

III. FINDINGS

From the interviews, relevant demographic information, such as position, gender, age, and working experience, were recorded, indicating there were 6 male and 4 female instructors, respectively, whose ages and working experiences ranged from 31 to 45 years and from 6 to 15 years, respectively. Table 1 shows the regarding information. Table 2 shows the information of the interviewees' current practices in teaching programming to novice students. The first question helped identify the type of techniques used by the instructors while the second question helped probe whether they had used any teaching aids. The answers revealed that all the instructors used discussion and problem solving as the main techniques in teaching programming. They also reported that the only teaching aid that they used was a module, which had been written by one of the lecturers of the center. Likewise, most instructors also reported that they occasionally used several reference books. In contrast, they reported that they had never used any type of courseware or learning portal in their teaching practice. Apparently, computer-based materials for the programming course have not thus far been provided to the center. The third question sought the instructors' opinions on the degree of difficulty in teaching the programming course. Overwhelmingly, all the instructors agreed that teaching programming course was difficult, citing the abstract, complex nature of the course and the lack of available teaching aids. In fact, they were compelled to spend a lot of time in preparing the teaching lessons and seek references to help them select the most appropriate teaching technique.

Table 1. The Demographic of the Programming Instructors

ID	Pseudo-name	Gender	Age (year)	Working experience (year)
R1	AAA	Female	38	6
R2	BBB	Male	42	13
R3	CCC	Male	40	15
R4	DDD	Male	40	7
R5	EEE	Male	45	8
R6	FFF	Male	39	9
R7	GGG	Female	31	5
R8	HHH	Female	40	10
R9	ZZZ	Female	37	6
R10	SSS	Male	43	6

The second set of questions was posed to the subjects to collect information pertaining to main teaching problems, the critical aspects of the problems, and the suggestions to improve the existing practice. Unanimously, the instructors reported that their students had problems in problem-solving and logical thinking, which they believed could be partly overcome by having a specific programming portal and greater collaboration among programming instructors and by introducing students to a problem-solving technique course prior to enrolling in the course. Table 3 summarizes the responses obtained after the end of the interview session. The third set of questions was posed to probe the instructors' understanding of the concept of computational thinking and their level of knowledge of computational thinking and of the relation between programming and computational thinking. The findings showed that almost all the instructors understood the concept of computational thinking and agreed that students' computational thinking skill could be enhanced through learning programming. Furthermore, they also agreed that the teaching technique that they used in teaching programming was suitable to enhance the computational thinking skill among novice students. Table 4 summarizes the responses obtained after the end of the interview session. Equally important, their opinions, perceptions, and ideas regarding the need for and the appropriateness of a learning portal had to be ascertained in this study. Through the interviews, they were asked about the specific characteristics and functionalities that they believed to be important. As expected, all the interviewees expressed their desire to have such a portal that could provide them learning materials that were easily accessible anywhere, anytime. Additionally, they stressed that contents uploaded to such a portal had to be carefully chosen and prepared to ensure their use could have a profound impact on student learning, notably on the development of students' computational thinking. Table 5 summarizes the opinions and suggestions of the instructors.



Table 2. Current Teaching Technique used by Programming Instructors in Teaching Programming to Novice Students

Question	Respondent										Supporting data resources
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	
What are the teaching techniques used by the instructor to teach programming? <i>i) Discussion</i>	√	√	√	√	√	√	√	√	√	√	10
<i>ii) Problem solving</i>	√	√	√	√	√	√	√	√	√	√	10
<i>iii)Others</i>	x	x	x	x	x	x	x	x	x	x	0
Does the instructor use any teaching aids in teaching programming? If the answer is 'Yes', state the learning aids.											
<i>i) Textbook</i>											
<i>ii) Module</i>	x	x	x	x	x	x	x	x	x	x	0
<i>iii)Computer Courseware</i>	√	√	√	√	√	√	√	√	√	√	10
<i>iv)Internet Resources</i>	x	x	x	x	x	x	x	x	x	x	0
<i>v)Reference books</i>	√	x	x	x	x	√	√	x	√	x	4
	√	√	√	√	√	√	√	√	√	√	10
Is the programming course difficult to teach? If the answer is 'Yes', why?											
<i>i) I have to make proper preparation in selecting the appropriate teaching techniques.</i>											
<i>ii) I have to cope with limited teaching aids.</i>	√	x	x	x	x	√	√	x	√	x	4
<i>iii) I believe programming is not easy for the students to learn.</i>	√	x	x	x	x	√	√	√	√	√	6
<i>iv) I believe some students have some misconceptions in certain topics.</i>	√	√	√	√	√	√	√	√	√	√	10
	√	x	x	x	x	√	√	x	√	x	4

A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

Table 3. Problem Encountered in the Current Approach

Question	Respondent										Supportive data resources
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	
<p>What are the main problems encountered in the current teaching approach?</p> <p><i>i) The students have problem in problem solving and logical and critical thinking.</i></p> <p><i>(ii) The students have misconceptions in certain topics.</i></p> <p><i>iii) The students have problem with the syntax of the programming language.</i></p>	√	√	√	√	√	√	√	√	√	√	10
	√	√	√	√	x	√	√	√	x	x	7
	√	x	x	x	x	√	√	√	√	√	6
<p>Which of the above problems deemed critical that entails the use of a new teaching tool?</p> <p><i>i) The students have problem in problem solving and logical and critical thinking.</i></p> <p><i>(ii) The students have misconceptions in certain topics.</i></p> <p><i>iii) The students have problem with the syntax of the programming language.</i></p>	√	√	√	√	√	√	√	√	√	√	10
	√	√	√	√	x	√	√	√	x	x	7
	√	x	x	x	x	√	√	√	√	√	6
<p>Do you have any suggestions to improve students' logical thinking and problem solving?</p> <p><i>i) Students should be introduced to a e problem solving technique course before learning programming.</i></p> <p><i>ii) Instructors should use certain teaching aids, such as a portal, that contain exercises that could stimulate students' logical and thinking problem-solving skill.</i></p>	√	√	√	√	√	√	√	√	√	√	10
	√	√	√	√	√	√	√	√	√	√	10

IV. DISCUSSION

The findings of this research highlight several important aspects of the teaching of programming to novice students. For example, a majority of the instructors relied on a discussion technique and problem-solving. They argued that such techniques were vital to help train novice programming students to generate ideas, the process of which involves logical thinking. In this regard, instructors need to be ingenious in using such a technique to encourage their students to solve problems in concert such that the latter can generate their own relevant ideas [16], [34] -[35]. In other words, instructors have to train their students to use the proper techniques in problem-solving. Together, the two techniques can be used to train and improve students' logical thinking and problem-solving skills. In addition, the findings showed that the instructors had to rely on a learning module and reference books in teaching the subject matter, with the former also being used by their students. Despite using the same module, the instructors emphasized that additional learning aids should be provided to their students to help the latter to practice and revise topics that they found challenging. As acknowledged, learning programming is not easy for most students, especially for novice students, as they have to not only learn and understand the concepts of programming but also master the problem-solving techniques and the specific language of programming. Arguably, such students will face a host of problems if they were to merely rely on existing learning materials. Obviously, the absence or the lack of new teaching and learning materials will aggravate the current predicament. To help mitigate such problems, new learning aids or applications have to be made available to students such that they use them outside the normal class hours irrespective of time and place. In this respect, a learning portal dedicated to the teaching and learning of programming is deemed important to provide students the opportunity to learn, practice, and discuss the subject matter more effectively. The use of such a learning system is in line with the findings of several studies, indicating that students' understanding of a particular course is dependent on the method of delivery used by instructors, computer-based instructions, and learning aids [44]. Addressing the second research question revealed that students learning programming would typically face two major problems. The first was that most of the students lacked the required set of skills, namely problem-solving, logical and critical thinking skills. In particular, most students, especially novice students, when given a programming problem, lacked the ability to think logically and critically to deal with such a problem, the ability of which has a huge impact on the development of their problem-solving skill. Therefore, students need to be trained through discussion and exercise along with the instructor in the classroom. Moreover, students also need to put some effort into carrying out extra problem-solving exercises outside the normal teaching hours, which can be achieved by using appropriate learning aids. The second problem was that the students had some difficulties in learning the syntax of the programming language. Essentially, every programming language has its own syntax, to which most students find difficult to grasp. Clearly, students have to do a lot of practice in order to master the syntax of the programming language. Admittedly, the lab sessions allocated to the course may not

be enough for the students to practice learning the language. Therefore, to familiarize with and master the language and its syntax, students have to do a lot of programming practices beyond the normal lab sessions. Thus, it becomes the imperative of instructors to provide learning materials, aids, and exercises (problems with solutions) that can help students to practice solving programming problems, the result of which can surely improve students' logical thinking. Preferably, an appropriate learning application (e.g., courseware or portal) hosting learning notes, teaching videos, exercises, and quizzes, among others, can be used to enable students to learn and apply the underlying principles of problem solving in a proper context that leads to the development of sound logical and critical thinking skills and computational thinking skill. The importance of such a learning portal as a learning tool has been emphasized by many scholars, with some arguing that there is sufficient evidence to support its use [45]-[49]. The third research question is concerned with the instructors' level of knowledge about CT skill. As revealed in this study, a majority of the instructors exhibited a sound understanding of the concept of CT in terms of decomposition, pattern, abstraction, algorithm, logical reasoning, and evaluation. Such a concept is consistent with the assertion made by [8], indicating that "there are various CT concepts that could be implemented in the classroom, such as logic, algorithm, decomposition, pattern, abstraction, and evaluation". Scrutinizing the CT concept will reveal that it is made up of several primary elements, namely algorithm, decomposition, abstraction, and evaluation, which are in fact the main concept of programming. Their sound understanding of the CT concept is not surprising, given that they had attended a workshop on CT skill organized by their schools. In the workshop, they learned the basic concepts of CT and the method of applying them in their teaching of programming. Essentially, solving programming problems involves a number of phases, namely analysis, design, implementation, and testing and validation phases. In the analysis phase, the input, process, and output have to be identified that entails the concept of abstraction to identify and extract relevant information to help define the main idea relating to a given problem. Such a phase also involves the concept of CT decomposition to decompose data, processes or problems into smaller, manageable parts. In the design phase, the sequencing of problem-solving steps is developed based on the given problem. Definitely, high logical reasoning is required in this phase to solve the given problem by visualizing it in terms of algorithms. According to [9], algorithms and logical thinking represent the concept of CT, with which the former helps create an ordered series of instructions for solving similar problems or for performing a task and the latter enables pupils to explain how such algorithms work, detect and correct errors, and predict outcomes.

A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

Table 4. Knowledge of Computational Thinking among Instructors

Question	Respondent										Supportive data resources
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	
Do you understand the concept of computational thinking?											
i) Yes	√	√	√	NS	√	√	√	√	√	√	9
If the answer is 'Yes', list down the main concepts of computational thinking.											
i)Decomposition											
ii)Pattern	√	√	√	NS	√	√	√	√	√	√	9
iii)Abstraction	√	√	√	NS	√	√	√	√	√	√	9
iv)Algorithm	√	√	√	NS	√	√	√	√	√	√	9
v)Logical reasoning	√	√	√	NS	√	√	√	√	√	√	9
vi)Evaluation	√	√	√	NS	√	√	√	√	√	√	9
Do you agree that students' computational thinking skill can be enhanced through learning programming?											
'Yes'	√	√	√	NS	√	√	√	√	√	√	9
If the answer is 'Yes', why?											
The concept of computational thinking is almost similar with the concept of programming.	√	√	√	NS	√	√	√	√	√	√	9
Which of the computational thinking concepts that you always come across in programming?											
i)Decomposition	√	√	√	NS	√	√	√	√	√	√	9
ii)Pattern	x	x	x	NS	x	x	x	x	x	x	0
iii)Abstraction	√	√	√	NS	√	√	√	√	√	√	9
iv)Algorithm	√	√	√	NS	√	√	√	√	√	√	9
v)Logical reasoning	√	√	√	NS	√	√	√	√	√	√	9
vi)Evaluation	√	√	√	NS	√	√	√	√	√	√	9
Do you think the teaching technique that you have applied in teaching programming is suitable to enhance the computational thinking skill among novice students?											
Yes	√	√	√	NS	√	√	√	√	√	√	9

NS – not sure



Table 5. Instructors' Perceptions of the Proposed Portal

Question	Respondent										Supportive data resources
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	
Do you agree that learning programming through a web portal can assist students to enhance their Computational thinking skill? Yes	X	√	√	NS	√	√	√	√	√	√	8
What type of an E-learning system that you think is suitable to be used? i) Portal ii) Web 2.0 applications (Facebook, Twitter etc.) iii) Online resources iv) Online game	x	√	√	NS	√	√	√	√	√	√	8
	√	x	x	NS	√	x	x	x	x	x	2
	x	x	x	x	x	x	x	x	x	x	0
	x	x	√	NS	√	√	√	√	√	√	7
What type of modules do you think are appropriate to be included in the proposed courseware/ portal? i) Concept description ii) Self-paced exercise (especially problem solving) iii) Online assessment iv) Notes	√	√	√	NS	√	√	√	√	√	√	9
	√	√	√	NS	√	√	√	√	√	√	9
	√	√	√	NS	√	√	√	√	√	√	9
	√	√	√	NS	√	√	√	√	√	√	9
What type of elements do you think should be integrated into the content delivery of the portal? i) Appropriate learning theory. ii) Motivational element. iii) Limit the time for students to complete tasks	√	√	√	NS	√	√	√	√	√	√	9
	√	√	√	NS	√	√	√	√	√	√	9
	√	√	√	NS	√	√	√	√	√	√	9
Do you foresee the proposed portal to be an important tool to assist novice students in enhancing their programming and computational thinking skills?	√	√	√	√	√	√	√	√	√	√	10
Could the proposed portal be pilot-tested for its effectiveness, which would involve your programming students? Yes	√	√	√	√	√	√	√	√	√	√	10



A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

In the implementation phase, coding, compilation, linking, and running and debugging of a program are carried out. Again, high logical thinking is required to develop a program and to identify and correct any errors in such a program, which in essence involves the concept of logical reasoning of CT. In the testing and validation phase, the developed program is tested using a number of inputs to ensure it will be free from errors and generate the desired outputs. All the activities carried out in this phase are in line with the evaluation concept of CT, which essentially “the process of ensuring that a solution, whether an algorithm, system, or process is good and fit the purpose” [9]. Clearly, the points raised in the above discussion might have prompted the instructors to state that their students' computational thinking skills, particularly that of novice students, could be enhanced through the learning of programming. Surely, to achieve such an enhancement, the instructors need to select and use the most effective teaching method in their teaching practice. As revealed in this study, it is worth to note that the process of solving problems and developing, testing and validating a program involves almost all the concepts of CT. The research findings also showed that a majority of the instructors agreed that the use of a learning portal could help improve their students' computational thinking skill. Admittedly, the practice of face-to-face teaching is indispensable; however, the use of relevant technologies, applications or tools can certainly help strengthen students' understanding of certain concepts learned and enable them to apply such concepts in solving a given problem. Obviously, solving programming problems with the use of the computer will entail the use of computational skill as well. Seemingly, such a contention concurs with the assertion made by Sakinah [23], who argues that, despite the fact that the best teaching model involves personal communication, the use of technology can help enhance student learning and make the presentation and delivery of learning materials more interactive. More specifically, she contends that online learning applications can generate a great deal on interest among students that can make students become more engaged, excited, and deeply motivated, ultimately leading to better learning performance. As such, instructors have to be diligent and resourceful to prepare and upload relevant learning materials to a portal, enabling students to gain access to such materials transcending time and geographical barriers. Currently, a host of computer-based learning applications are available, such as standalone courseware, e-learning portals, online games, and Web 2.0 applications, including Twitter, Facebook, and Instagram, among others, can be used for teaching and learning purposes. Interestingly, many studies have shown that a majority of instructors prefer to use web-based portals as the learning application or system to teach programming to novice students. Such a preference is to be expected due to the fact that a web-based portal serves as a one-stop center at which students can gain access anywhere, anytime. Clearly, using such a portal, students are able to perform many tasks online, such as downloading lecture notes and materials, submitting their lab assignments, and communicating with their lecturers and friends. In fact, a web-based portal offers a feature for students to communicate with other online users, such as their lecturers, classmates and college friends. They can also view who is online and have conversations with him or her to discuss virtually anything pertaining to either academic or non-academic matters.

Additionally, they can pose questions to their lecturer too if the latter is available online. In view of the many educational benefits offered, it is hardly surprising to note that the number of web-based learning portals has increased exponentially over the recent years, making them commonplace in many schools, colleges, and universities. Such pervasive use can be explained by the contention made by Morin, Thomas, and Saade [50], who stress that “some level of web-based learning has now become the norm rather than the exception. Even classes that principally use a traditional classroom, lecture approach to delivery often integrate some level of web-based learning via course management systems, often referred to as blended learning”. Another feedback is concerning the elements that should be integrated into the contents hosted by such a portal. Clearly, the development and delivery of such contents be guided by strong, sound theoretical principles to ensure learning through the portal will be educationally efficacious [51]. To date, several contemporary learning theories, such as multimedia learning theory, cognitivism, and constructivism, have been widely used for such a purpose. Based on a critical review of the current literature, the proposed portal will integrate the principles and precepts of the cognitivist theory into the learning contents. From the cognitive perspective, the use of such principles can help students to learn by actively processing information in order to create new knowledge. In a nutshell, the cognitive theory focuses more on students' mental activities and processes that lead to a better understanding of what has been learned [52]. In line with the cognitivist principles, the proposed portal will include activities entailing learners to apply, analyze, synthesize, and evaluate information that ultimately leads to better higher-order thinking skills (HOTS). In addition, motivational elements need to be taken into account in any courseware development, which according to some scholars [5] [54] [56], including Popovich [53], can help persistently engage students in learning and attain better learning performance. Furthermore, the proposed portal should also include a real-time monitoring feature to track and monitor students' online activities, such as online discussions, chats, quizzes, and any forms of communication. It needs to be strongly emphasized that the proper design of a learning portal and the use of appropriate learning contents and activities play a crucial in the acquisition of problem-solving skills among students, especially among novice students. Such an emphasis is in line with the findings of previous studies showing that the activities and resources used in a particular course have a significant impact on the acquisition of various higher-order learning skills, including problem-solving, among students [57] – [60] students, collectively contributing to the development of sound computational thinking skills.

To help realize such type of learning, the researchers propose the following components to be used in the development of the portal: (i) concept description, (ii) the provision of examples, (iii) tutorials, and (iv) self-paced exercises. From the educational standpoint, the proposed portal can serve as an important tool to assist novice students to enhance their programming skills and computational thinking skills through the learning of programming. An investigation on the effectiveness of this portal



can help provide the relevant answer to the fourth research question that seeks to examine the instructors' perceptions on the development and implementation of the proposed web-based programming learning portal.

V. RECOMMENDATION

As a whole, the findings of the study based on the input from the instructors themselves provide several important insights. Current teaching practices seem not in harmony with today's teaching demands, learning tools and applications are seldom used at best, or are not available at worst, and in cases where these tools are available they lack proper instructional design elements. Given these drawbacks, it is naturally understandable that the instructors who were interviewed had cautioned the prevailing problems and emphasized immediate solutions such as using web based applications, tutorials, and other forms of materials. Overall, the web-based programming learning portal has received positive perceptions and feedback regarding all the recommended features, which will be used in the development of such a learning portal in the next stage of the research. Specifically, the learning portal, called My-ProgCT, will be designed and developed based on the proposed Instructional Design (ID) model of the content delivery of the programming portal My-ProgCT as shown in Fig. 1. The ID model consists of the following phases.

(i) Analysis

From the literature review, an appropriate learning theory and motivation model that are suitable to be embedded in the content delivery of the programming portal My-ProgCT namely Cognitive Learning Theory and Motivation ARCS model.

(ii) Design

The content delivery of the programming portal will be designed based on the Cognitive Learning Theory and the elements of ARCS model. On top of that the Inductive Approach will also be used in this phase.

(iii) Development

The Programming Portal – My-ProgCT will be consisted of four (4) modules namely Concept Description, Tutorial, Example and Self-paced Exercise. The modules will be able to perform the following tasks: (i) to clearly show and explain the concept of programming and (ii) to show how to solve problem-solving tasks. In addition the elements of interactive multimedia namely text, graphic, animation, audio and video will also be embedded in the modules to make learning through the portal highly interactive, responsive, and dynamic.

(iv) Outcome

The novice students' CT skills which consist of Problem Solving and Critical Thinking skills will be enhanced through the using of the Programming Portal My-ProgCT.

VI. CONCLUSION

As demonstrated in this study, the current research findings concur with previous findings, strongly reinforcing the contention that the CT skills of students, notably novice students, can be improved through the learning of

programming, which seems possible from the theoretical perspective. Nevertheless, from the practical standpoint, such improvement may be difficult to accomplish, given the prevailing pedagogical teaching practices that primarily rely on the conventional teaching method used by instructors, namely discussion and problem-solving technique. Admittedly, such a method is not highly effective as the time allocated for discussion and problem-solving exercises is very limited, putting students with learning problems in a dilemma when they leave their class. Coupled with a lack of effective learning aids, such a predicament becomes more acute. In view of such a problematic situation, therefore, a new teaching aid that can be used anywhere and anytime is needed. To help mitigate such problems, new learning aids or applications have to be made available to students such that they use them outside the normal class hours irrespective of time and place. Preferably, such learning aids should serve as a platform that not only hosts learning materials but also provides a space that facilitates collaboration among students, with the latter providing students with cognitive scaffolds to solve problems, as viewed from the social learning perspective. In this respect, a learning portal dedicated to the teaching and learning of programming is deemed important to provide students the opportunity to learn, practice, and discuss the subject matter more effectively. On a final note, with more efforts put in, the Programming Portal My-ProgCT can become a potent learning tool with which instructors can effectively use to teach programming to novice students, the impact of which will help enhance not only the latter's programming skill but also their computational thinking skills.

VII. ACKNOWLEDGMENT

The authors wish to acknowledge the support of the Research and Management Centre, Universiti Pendidikan Sultan Idris with a research grant (GRU - 2018-0044-107-01) that helped fund the study.

REFERENCES

1. 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. 1–17. doi:10.1111/jcal.12099, 2015.
2. S. I. Malik, & J. Coldwell-Neilson, "Impact of a New Teaching and Learning Approach in an Introductory Programming Course," *Journal of Educational Computing Research*, vol 0, no 0, pp. 1–31. doi:10.1177/0735633116685852, 2017.
3. M. H. Moshref Javadi, M. Ghandehari & V. Hamidi Pouyandeh, "Locating of Bicycle Stations in the City of Isfahan Using Mathematical Programming and Multi-Criteria," *International Journal of Academic Research in Progressive Education and Development*, vol 6, no 1, 2013.
4. N. Park, "Development of Computer Education Program Using LOGO Programming and Fractals Learning for Enhancing Creativity: Focus on Creative Problem-Solving," *International Journal of u- and e- Service, Science and Technology*, vol 9, no 2, pp. 121–126, 2016.
5. J. Wing, "Computational thinking and thinking about computing," *Philosophical Transactions of The Royal Society* (July), pp. 3717–3725. doi:10.1109/IPDPS.2008.4536091, 2008.
6. J. Voogt, P. Fisser, J. Good, P. Mishra & A. Yadav, "Computational thinking in compulsory education: Towards an agenda for research and practice," *Education and Information Technologies*, vol 20, no 4, pp. 715–728, doi:10.1007/s10639-015-9412-6, 2015.



A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

7. 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, New York: ACM, 2015.
8. Barefootcas, "Computational Thinking," Retrieved from <https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/computational-thinking/>, 2014, July.
9. MDEC, "Computational Thinking and Computer Science Teaching Certificate Programme for Educator," Cyberjaya, 2018.
10. S. I. Swaid, "Bringing computational thinking to STEM education," *Procedia Manufacturing* vol3, pp. 3657–3662. doi:10.1016/j.promfg.2015.07.761, 2015.
11. C. Wilson, & M. Guzdial, "How to make progress in computing Education," *Communications of the ACM*, vol 53, no 5, pp. 35–37, 2010.
12. 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, DOI 10.1186/s41239-017-0080-z, 2017.
13. K. Cagin, K. Mary, B. Liz & M. Lachlan, "A serious game for developing computational thinking and learning introductory computer programming," *Procedia - Social and Behavioral Sciences*, vol 47, pp.1991 – 1999, 2012.
14. 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.
15. Y.B. Kafai, "From computational thinking to computational participation in K-12 education," *Communications of the ACM*, pp. 26–27. doi: 10.1145/2955114, 2016.
16. T. Djambong & V. Freiman, "Task-Based Assessment of Students' Computational Thinking Skills Developed Through Visual Programming or Tangible Coding Environments," *13th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2016)*, pp. 41–52, 2016.
17. H. S. Masharah, "Model Kompetensi Guru Pengaturcaraan," PhD Thesis – unpublsh, Universiti Pendidikan Sultan Idris, Perak, Malaysia, 2017.
18. C. J. Costa & Cordeiro, 2012; Costa, C. J., Aparicio, M., & Cordeiro, C. (2012). *A solution to support student learning of programming*. Paper presented at the Proceedings of the Workshop on Open Source and Design of Communication.
19. M. Havenga, E. Mentz, E., B. Breed, D. Govender, I. Govender, F. Dignum, & V. Dignum, "A case study regarding teachers' problem-solving activities and approaches towards computer programming in diverse learning environment," Paper presented at the 3rd International Conference on Society and Information Technologies ICSIT, Orlando (USA), 25-28 March, 2012.
20. K. C. Kessler & J. R. Anderson, "Learning flow of control: Recursive and iterative procedures," *Human-Computer Interaction*, vol 2, no 2, pp. 135-166, 1986.
21. M. R. Siti Nurulain (2015) *A Metacognitive Support Environment For Novice Programmer Using Semantic Web*, PhD Thesis – unpublsh, Universiti Of Malaya, Kuala Lumpur, 2015.
22. I. Miliszewska & G. Tan, "Befriending computer programming: A proposed approach to teaching introductory programming," *Journal of Issues in Informing Science and Information Technology*, vol 4, pp. 277-289, 2007.
23. M. Y. Siti Sakinah, "Keberkesanan Penggunaan Alat Bantu Mengajar Interaktif bagi Meningkatkan Kefahaman Konsep Pengaturcaraan dalam Kalangan Pelajar," Masters Dissertation – unpublsh, Universiti Pendidikan Sultan Idris, Perak, Malaysia, 2017.
24. P. B. Henderson, T. J. Cortina, J. M. Wing, "Computational Thinking," *ACM SIGCSE Bulletin*, vol 39, pp. 195–196, 1987.
25. M. C. Linn & M. J. Clancy, "The case for case studies of programming problems," *Communications of the ACM*, vol 35, no 3, pp. 121-132, 1992.
26. R. Poli & J. Koza, *Genetic Programming*, pp. 48-60, Berlin: Springer, 2014.
27. E. Soloway & J. C. Spohrer, *Studying the novice programmer*: New York, NY: Psychology Press, 2013.
28. R. E. Mayer, *Learning and instruction*, Upper Saddle River, NJ: Prentice Hall, 2003.
29. S. I. Malik & J. Coldwell-Neilson, "Impact of a New Teaching and Learning Approach in an Introductory Programming Course," *Journal of Educational Computing Research* pp. 1–31. doi:10.1177/0735633116685852, 2017.
30. S. M. Biju, "Difficulties in understanding object oriented programming concepts," In *Innovations and Advances in Computer, Information, Systems Sciences, and Engineering*, pp. 319-326, Dubai: Springer, 2013.
31. R. Killen, *Programming and Assessment for Quality Teaching and Learning* (1st ed.), Southbank, Vic., Thomson Social Science Press, 2005.
32. M. Saeli, J. Perrenet, W. M. G. Jochems & B. Zwaneveld, "Programming: Teachers and pedagogical content knowledge in the Netherlands," *Informatics in Education*, vol 11, no 1, pp. 81-114, 2012.
33. O. Hazzan, T. Lapidot, & N. Ragonis, *Guide to teaching computer science: An activity-based approach*, DOI 10.1007/978-0-85729-443-2_8, London: Springer-Verlag., 2015.
34. M. P. Uysal, "Improving First Computer Programming Experiences: The Case of Adapting a Web-Supported and Well-Structured Problem-Solving Method to a Traditional Course," *Contemporary Educational Technology*, vol 5, no 3, 2014.
35. G. Mason, T. R. Shuman & K. E. Cook, "Comparing the effectiveness of an inverted classroom to a traditional classroom in an upper-division engineering course," *IEEE Transactions on Education*, vol 56, no 4, pp. 430–435, 2012.
36. S. Craciunas & I. Elsek, *The standard model of an e-learning platform*. Bucharest, Romania, 2009.
37. S. Sharma, & S. Gulsecen, S. "Using web-based learning resources for e-learning Exploratory Study," *Second International Conference on Innovations in Learning for the Future, e-Learning Conference Proceedings*, pp. 517-524, 2008.
38. 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.
39. KPM, "Kurikulum Standard Sekolah Rendah KSSR," B. P. Kurikulum, Ed., 2016.
40. C. C. o. H. Capital, J. P. K. d. KPTM, & J. P. Teknikal, "Detailed Concept Paper: Study on the Development of ICT Curriculum Standards Coordination in Primary and Secondary Schools in Malaysia," ed: Ministry of Higher Education Malaysia, 2013.
41. K. P. Malaysia, "Pelan Pembangunan Pendidikan Malaysia 2013-2025," Kementerian Pendidikan Malaysia, 2012.
42. S. Aslina, "Developing Students' Computational Thinking Skill Through Cooperative Learning Based On Hands-On, Inquiry-Based, And Student-Centric Learning Approaches," Conference: 2nd International Teacher Education Conference On Teaching Practice (ITECTP 2018), Kuala Lumpur, 2018.
43. C. R. Kothari, *Research methodology* (2nd ed.). New Delhi: New Age International Publishers, 2004.
44. M. Nurul Nisha, "Pembangunan Modul ICTL Berasaskan Pendekatan Pengajaran Langsung Dan Teori Konstruktivisme Untuk Pelajar Orang Asli," Masters' Thesis - unpublsh, Universiti Pendidikan Sultan Idris. Tg. Malim Perak, 2012.
45. M. Omi, A. Michal & B. Mordechai, "Learning computer science concepts with scratch," In *Proceedings of the Sixth international workshop on Computing education research (ICER '10)*. pp. 69-76, 2010.
46. M. Reginamary, S. H. Hin & A. C. Koo, "Multimedia learning object to build cognitive understanding in learning introductory programming," In *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM '09)*, pp. 396-400, 2009.
47. A. Fatima & E. Ashraf, "Learn-DG: Java learning system using dependence graphs," In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services (iiWAS '09)*, pp. 633-637, 2009.
48. P. Ana, L. Ambrósio & M. C. Fábio, "Evaluating the impact of PBL and tablet PCs in an algorithms and computer programming course," In *Proceedings of the 41st ACM technical symposium on Computer science education (SIGCSE '10)*, pp. 495-499, 2010.
49. F. Georgios & S. Kiriaki, "Influence of the familiarization with "scratch" on future teachers' opinions and attitudes about programming and ICT in education," *SIGCSE Bull.*41, pp. 258-262, 2009.
50. D. Morin, J. D. E. Thomas & R. G. Saadé, "Fostering problem-solving in a virtual environment," *Journal of Information Technology Education: Research*, vol 14, pp. 339-362. Available: <http://www.jitee.org/documents/Vol14/JITEv14ResearchP339-362Morin2031.pdf>. 2015.
51. R. M. Gagne, *Instructional technology: Foundations*, Florida State University, London, Hillsdale New Jersey: Lawrence Erlbaum Associates Publishers, 1987.



52. M. Y. Siti Ilyana MohdYusof, "The Development And Evaluation Of An Independent Model Based On Connectivism Theory And Web 2.0," PhD Thesis – unpublsh, Universiti Pendidikan Sultan Idris, Perak, Malaysia, 2017.
53. N. G. Popovich & S. Wongwivatthanannukit, "Applying the ARCS Model of motivational design to pharmaceutical education", *American Journal of Pharmaceutical Education*, vol 64, pp. 188-196, 2000.
54. J. M. Keller, "Motivation and instructional design: A theoretical perspective", *Journal of Instructional Development*, vol 2, no 4, pp. 26-34, 1979.
55. J. M. Keller, *Motivational design of instruction*, In. Reigeluth, C. M. (Ed.). Instructional-design theories and models: An overview of the current status. Hillsdale, New Jersey, Lawrence Erlbaum Associates, Inc., 1983, 383-434.
56. L. Y. Rahmah & T. P. Tengku Nazatul Shim, "Reading Activities Using the Scaffolding in MEL-SindD for Down Syndrome Children," In the Asia Pacific International Conference, pp: 121 – 128, 2011.
57. D. Morin, J. D. E. Thomas & S. Ly, "The effect of delivery method on persistence, performance and perceptions," *International Journal of Excellence in Education*, vol 6, no 2, pp. 1-10, 2014.
58. D. Morin, J. D. E. Thomas & R. G. Saadé, "Activities and resources in online learning: From a critical thinking view," In *Proceedings of Informing Science & IT Education Conference (InSITE) 2012* pp. 597-602, 2012.
59. R. G. Saadé, D. Morin & J. D. E. Thomas, "Critical thinking in e-learning environments," *Computers in Human Behavior*, vol 28, no 5, pp. 1608-1617, 2012.
60. J. D. E. Thomas & D. Morin, "Technological Supports for Onsite and Distance Education and Students' Perceptions of Acquisition of Thinking and Team-Building Skills, *International Journal of Distance Education Technologies*, vol 8, no 2, pp. 1-13, 2010.

AUTHORS PROFILE



Dr. Nor Hasbiah Ubaidullah is working as an Associate Professor at Universiti Pendidikan Sultan Idris. She received degree in Computer Science from Universiti Kebangsaan Malaysia and master of Science in Information System from University of Salford. She obtained her PhD degree in Information Technology from

Universiti Kebangsaan Malaysia. She has been teaching in higher education institutions since 1991 in several subjects such as Programming, Courseware Engineering and Information Systems Development. Her research areas include Data Management, Educational Software and Computer Science Education.



Dr. Jamilah Hamid is a Senior Lecturer in Faculty of Art, Computing and Creative Industry. She holds a Bachelor of Science in Education (Hons.) majoring in Mathematics and Master of Science (Information Technology). She obtained her PhD in Internet and Web Computing from Universiti Pendidikan Sultan Idris, Malaysia She has been

teaching in high education institutions since 1991 in several subjects such as Programming, Methodology in Courseware Development, Educational Technology and Teaching Methodology. Her research interests are on Virtual Reality, Educational Technology, Educational Software and Teaching Methodology in Programming. Currently, she is attached to Universiti Pendidikan Sultan Idris and teaching Methodology in Courseware Development.

A Web-Based Learning Programming Portal: Do Instructors Need It to Enhance Novice Students' Computational Thinking Skill?

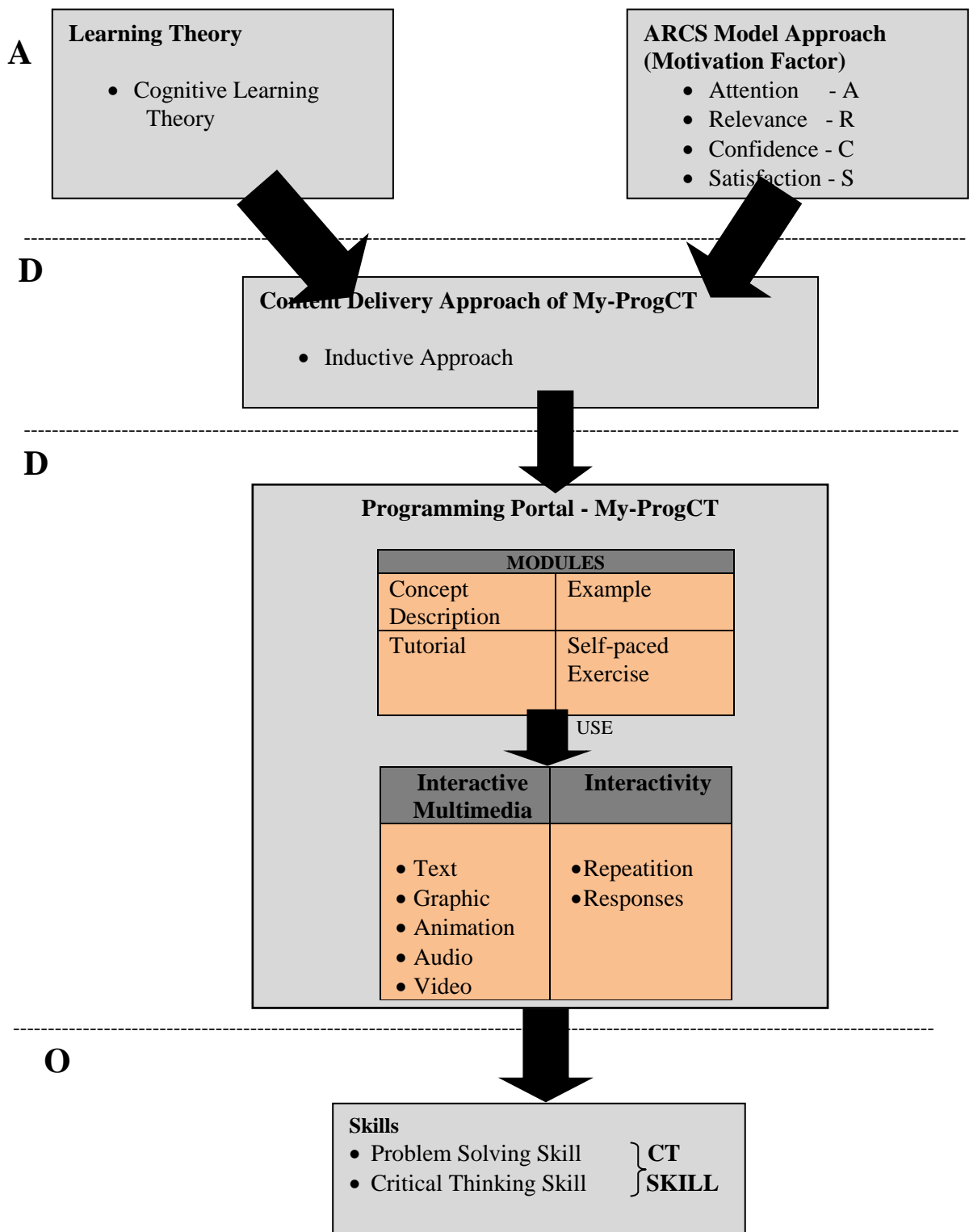


Fig. 1. The propose Instructional Design Model of the content delivery of the programming portal My-ProgCT