Designing an Electronic Course and Its Impact on Developing University Students' Computational Thinking Skills

Majed bin Maili bin Mohammad Al-Zahrani †

Umm Al-Qura University, Makkah, Makkah, SA

Summary
The study investigated the effects of differences in some electronic course designs on university students' Computational Thinking Skills (CTS). Towards this end, the researcher adopted the experimental research design of a quasi-experimental of two experimental groups. The first group was taught an e-course designed in a sequential pattern, and the other group's course was designed according to the holistic model. A CTSs test was prepared to collect the relevant data, and the data were analyzed statistically using these tests- Pearson correlation Mann Whitney and Alpha Cronbach. Results revealed statistically-significant differences at the level α=0.05 between the mean scores of the first and second experimental groups in favor of the latter in the CTS test. The findings gave ground to put forward some salient recommendations, including the need to expand computational thinking in universities' educational process. It also recommends urging faculty members to enhance e-courses in the educational process and provide technical support to students and faculty members.

Key words:
Electronic Course and Its Impact, Computational Thinking Skills

1. Introduction

Numerous educational systems in advanced countries have included Computational Thinking Skills (CTSs) in the educational curricula at the multiple stages of public education. In the United Kingdom, the Ministry of Education has developed legal frameworks that obligate curriculum developers to include CTSs in educational curricula in various educational stages. In the United States of America, many professional, academic and industrial organizations have been concerned with computational thinking, whether at research and studies or education and teaching practices. In the undergraduate and pre-university education stages, the Council of Computer and Communication Sciences held a series of workshops on computational thinking, focusing on identifying the basic concepts of computer science that can be taught at the primary level [1].

Given the rapid developments in the educational curricula in Saudi Arabia, the Computer Curriculum Document (2013) intends to develop and design computer curricula according to international standards to include several skills – the most important are computer thinking skills in the topics presented to students of general education [2].

Computational thinking is used in many disciplines, such as science and engineering, using modeling, simulation, data mining, machine learning, and Big Data Analysis. Employing computer thinking skills in education does not necessarily mean the use of devices. Instead, it rather means using the best problem-solving strategies in addition to mathematical and algorithmic thinking to train students to solve problems in innovative ways based on a scientific approach [3].

According to [2], the International Society for Educational Technologies (ISTE) and the Computer Science Teachers Foundation (CSTA) agree on the view of computational thinking in terms of the concept and skills that it seeks to impart to learners. This includes problem formulation, identification, knowledge of its components, and information gathering that has a close relationship with the problem, its analysis and organization, neglecting information that does not constitute a solution to the problem or part of it, knowing patterns and models that help us solve the problem, and representing what has been reached in the form of algorithms that help us get a solution to the problem, testing those solutions to ensure the correctness of the solutions that have been Reaching them and their feasibility in solving the problem, then reproducing the solutions that were reached in solving similar problems in different situations.

A sequence of studies focused on computational thinking indicating an increasing interest from the public and private educational institutions, whether in education systems in developed countries or developing countries. This is because it provides learners with the necessary skills that help them to accomplish their daily tasks in a better way and provide them with the skills that help them engage in The labor market in their immediate future [4].

Computational thinking, according to [5], has the following characteristics:

- Computational thinking is a concept based on computer science concepts with a high degree of abstraction, not just programming.
- It equips learners with essential skills to live in a society of continuous development, rather than skills

Manuscript received December 5, 2021
Manuscript revised December 20, 2021
https://doi.org/10.22937/IJCSNS.2022.22.1.3

characterized by monotony and lack of adaptation to developments.

- It simulates how humans think to solve problems; its goal is not to enable human to think and deal with issues the same as the machine; humans are able to innovate, be intelligent, and able to adopt to problems.
- It combines mathematical and engineering thinking because computer science depends in many of its operations on mathematics and engineering in building systems that are intelligent and interact with humans.
- It seeks to form a scientific methodology in solving and managing the problems facing the learners in their daily life in a manner that does not affect their interaction and communication with others.
- CTSs can be applied with all members of society of all ages, specializations, and practical inclinations as long as an individual has a desire to learn. Therefore it is the current reality and the new philosophy of man in the twenty-first century.

The application of computational thinking in education in the primary stage started in 2009 by the Association of Computer Science Teachers and the International Association for Educational Technology. The project focused on finding definitions, projects and educational curricula suitable for computational thinking. CTSs can be taught in education in two ways: either as a course given to computer sciences students in which computer science concepts and skills are introduced or applying CTSs as an educational strategy concerned with integrating its concepts and skills in various academic subjects according to the nature of their content [1].

[6] maintained that in light of technological developments, the teacher should be familiar with the three components of knowledge. The first component is knowing and deepening the educational content related to academic specialization. The second component is the scholarly knowledge of professional aspects of teaching methods and strategies and other expertise that helps teachers perform their mission. The third component is technical knowledge, which requires the teacher to be constantly aware of the latest technologies used in his specialization and how to take advantage of them and employ them in the educational process.

In partnership with Google, a group of researchers at the University of Canterbury in New Zealand launched a project to produce educational lessons about computational thinking in pre-university education. The project, known as Computer Science Unplugged, aimed to support educational content with many educational activities that help learners acquire CTSs using simple tools available in the learner's environment. It also aims to helping the learners to develop essential skills if they desire to continue their education in the field of computers [1].

When thinking about teaching computer science in schools, teachers should introduce a specific sequence of essential computer science topics to students. Undoubtedly, computer science requires knowledge of mathematics, discrete mathematics, linear algebra, and problem-solving. Nevertheless, building students' capacity for deep learning computer science skills is the main reason students succeed in computer science and continue their studies after school [3].

Many teaching methods can be used in teaching computational thinking, including educational scaffolding. [7] accentuated training a group of middle school students on computational thinking and its concepts and skills that relate to problem-solving and learning about the design pattern of algorithms. The researcher designed an educational model based on solving problems and educational scaffolding as an educational method to provide students with the basics of programming, and the results showed a positive attitude towards computers and a positive effect of computational thinking.

[8] postulated that learning based on educational scaffolding speeds up learning of CTSs for pre-university learners.

Computational thinking has some skills that help in solving problems. According to [3] and [2], those skills are the following:

Skill 1: Problem Decomposition

Analyzing the problem into small parts makes it easier for learners to deal with it, helps them learn, and motivates them to deal with it and solve it, unlike complex or complex problems. This skill can be applied in various academic subjects and is not limited to computer science.

Skill 2: Pattern Recognition

This skill helps to know the similarities and differences between the small parts of the problem or the pattern of repetition of this problem. This skill allows learners to write algorithms efficiently and not repeat programming commands if the solution is implemented on a specific programming language.

Skill 3: Abstraction

Abstraction is one of the highest and most important levels of CTSs. It is the process of examining small parts of the problem and omitting details that are not closely related to solving the problem; focusing on the important details that constitute an entrance to solving the problem and thus helps to focus on them in the steps of solving the problem.

Skill 4: Algorithm design

In this skill, the solution to the problem is crystallized in the form of successive steps designed in light of the previous steps and are represented either using flow charts or using algorithms or semi-formal code (Pseudo code).

2. Problem Statement

Some previous studies have ascertained that teaching CTSs is not devoid of difficulties and challenges. The most significant obstacles are the teacher's weak academic preparation and lack of familiarity with CTSs and how to use them in teaching educational content. In a similar vein, resistance to change when using any technology or educational practice teachers are unfamiliar with is another challenge. Moreover, weak infrastructure is an obstacle to providing tools or techniques that help learn CTSs [2].

[9] believes that computational thinking helps to decompose problems into small parts. This helps not only in programming but also in understanding the components of the problem and how to deal with it, identifying patterns, extrapolating solutions, and ensuring their feasibility. Many specialists worldwide have claimed that computational thinking is an essential element when teaching programming to learners. Hence, the goal is not to learn programming as a goal in itself, but rather should include dealing with problems and creating algorithms of solution and how to generalize the solutions that have been reached to similar issues. A tool that promotes learners' creativity, develops thinking skills, and enhances collaborative learning skills, not just writing programming codes.

Learning and developing CTSs effectively contribute to students learning skills that help them face many situations they find in life and enable them to have skills in dealing with computers and how to harness it in the service of humans [10].

3. Research Questions

The study attempts to answer the following two questions:

Q1: What are the necessary CTSs for Umm Al-Qura University students?

Q2: What is the impact of the two styles of electronic course design (holistic & sequential) on developing CTSs?

4. Objectives:

The study intends to achieve the following objectives:

- Finding a list of the CTSs to be learned for higher education students.
- Recognizing the effect of the sequential and holistic design patterns of an e-course on developing skills.

5. Method

Research Design

The researcher adopted the experimental method with a quasi-experimental design for two experimental groups: the first group studied an e-course designed in the holistic style, and the second studied an e-course designed in a sequential style. The dependent variable was computer thinking skills.

Population

The research population consists of all students of Umm Al-Qura University (*N*=1522).

The Computational Thinking Skills (CTSs)

The researcher prepared a list of CTSs according to the following steps:

- A review of some studies and research that dealt with CTSs.
- Examining the experiences of some international institutions in teaching computational thinking.
- Write a list of CTSs and present it to specialists in educational technology and computers for arbitration.

Validity and reliability of the tool

The researcher handed in the list of CTSs to some experts to check its validity. As for the reliability of the list, it was calculated by using Coopers' equation, according to which the observers' agreement coefficient was 80%, which is adequate to carry on with the tool.

$$\text{Coeficient of agreement} = \frac{\text{number of times agreed}}{\text{number of times greed} + \text{number of times difference}} \times 100$$

Computational thinking test

The computational thinking test was designed with 15 multiple-choice questions. Each question has three alternative answers with only one correct answer. In addition, three essay questions were presented to the student as problems, and the student applied the CTSs to them as required in each question. Thus, the total of the test items is (18) items.

Validity and reliability of the test

The validity of the test means the ability of the test to measure what it was designed to measure, as mentioned by [11]. The validity of the test was estimated in the current research in two ways:

- ***Face validity***

The face (apparent) validity of the test relates to the extent to which the test reflects the objectives of the course to be measured, also called 'content validity'. The initial version of the test was presented to some specialists, and the test consisted of 18 items, of which 15 items were of the type of multiple-choice, and three essay questions.

- ***Internal validity***

Internal validity is concerned with how the test represents the CTSs designed to measure them, ascertained by measuring the extent to which the test items are related to the levels of the objectives to be measured.

(remembering, understanding, applying, analyzing), on those topics, the number of test items that measure those goals, and their relative weights. Table 1 shows the specifications of the computational thinking test:

Table 1. *Specifications of the Computational Thinking Test*

Topics	Remember		Comprehension		Application		Analysis		Wt.	# Qs
	objective	Q	objective	Q	objective	Q	objective	Q		
Introductory Comp. thinking	1	1	-	-	3	3	-	-	27%	4
Problem Decompose	-	-	1	1	3	3	1	1	33%	5
Pattern Recognition	-	-	2	2	-	-	-	-	13%	2
Abstraction	-	-	1	1	-	-	-	-	7%	1
Algorithm	-	-	-	-	3	3	-	-	20%	3
Total	1	1	4	4	9	9	1	1	-	15
%	6.7%	6.7%	26.6%	26.7%	60%	60%	6.7%	6.7%	100%	-

Reliability of the Test

Reliability measures the extent to which the test yields the same or close scores when applied more than once and in the same circumstances [11]. The reliability of the computational thinking test was calculated by test-retest, and the value of Cronbach's Alpha was 0.847, which means that the test has a high degree of stability.

Computational thinking pre-test

To ensure that the two research groups were equal in their CTSs, the researcher distributed a paper copy of the test to the two research groups in the second meeting of the first week of the experiment. Then the researcher corrected the test and then analyzed the scores using the SPSS program, and the results were as follows:

Table 2. *Results of the pre-test*

Test	Sample	N	Rank means	Z	Sig.	Mann Whitney
Computational thinking Test	Experimental 1	15	14.84	0.745	0.456	101.500
	Experimental 2	15	17.23			

Z at 0.01 = 2.58
Z at 0.05 = 1.96

In Table 2, the calculated “z” value (0.745) is less than the tabular “z” value at the level of significance (0.05), and this indicates that there are no statistically significant differences between the two research groups. Therefore the condition of parity between the two research groups is fulfilled.

Tool post application

The test was distributed to the two research groups, then corrected, scores were monitored and analyzed using the SPSS program to test the validity of the hypothesis.

6. Results

Hypothesis

There are no statistically significant differences between the mean scores of the students of the first experimental group and students of the second experimental group in the post application of the computer thinking skills test. To verify this hypothesis, the researcher applied the Mann-Whitney test for independent samples that do not meet the conditions of the t-test, either because of the lack of moderation or the poor representation of the sample for the research population. The significance of the differences between the means is outlined in Table 3 below.

Table 3. *Students' results in the the computational thinking Post-Tes*

Test	Sample	N	Z	Sig.	Mann Whitney	Rank means	Coeff.	Size effect
Computational thinking test	Experiment 1	15	2.045	0.01	63.500	12.23	0.436	moderate
	Experiment 2	15				18.77		

Z at 0.01 = 2.58
Z at 0.05 = 1.96

The data displayed in the table shows the Z value is 2.045, which is greater than the tabulated Z at 0.05. This indicates statistically significant differences in favor of the first experimental group that studied using an e-course in the holistic style because its ordinal average was (18.77), more important than the means of the first group that studied through an e-course designed sequentially. On this basis, the null hypothesis was rejected, and the alternative hypothesis was accepted. There are statistically significant differences between the mean scores of the students of the first experimental group and the students of the second group in the post-application of the computational thinking test in favor of the first group that underwent the sequential pattern. To determine the size of this effect, the researcher used the binary correlation coefficient for ranks (see Table 3). As displayed in the table, the value of the coefficient is 0.436, which is, according to [11], a medium effect. This indicates that the holistic module was effective in developing CTSs. Perhaps, this effectiveness is attributed to the fact that the holistic style in presenting CTSs gives information in an integrated manner, which helps students understand and master the educational content.

References

[1] Al-Taher, M. & Fershishi, A. (2017). *Computational thinking approach, Binary Relationship Methodology, and Algorithms*. A proceeding of the 11th International Conference on Computer Science and Engineering. August 2017, Sudan, Khartoum.
[2] Al-Juwaid, M. & Al-Obeikan, R. (2018 AD). Training needs of computer teachers to use and teach computer thinking skills. *International Journal of Educational Research, 42* (3), UAE University. The United Arab Emirates.
[3] Awadh, I.A. (2018). *Computational thinking*. A scientific article published on the Shams platform.
[4] Basogain, X., Olabe, M. A., C., O. J., Ramírez, R., & García, J. (2016). PC-01: Introduction to Computational Thinking. Educational Technology in Primary and Secondary Education. In F. J. García-Peñalvo & J. A. Mendes (Eds.), XVIII Simposio Internacional de Informática Educativa, SIIE 2016 (pp. 191-195). Salamanca, España: Ediciones Universidad de Salamanca.
[5] Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.
[6] Koehler, M. J., Mishra, P., Kereluik, K., Shin, T. S., & Graham, C. R. (2014). The technological pedagogical content knowledge framework. In *Handbook of research on educational communications and technology* (pp. 101-111). Springer New York.
[7] Webb, H. C. (2013). *Injecting computational thinking into computing activities for middle school girls* (PhD dissertation). Pennsylvania State University.
[8] Touretzky, D. S., Marghitu, D., Ludi, S., Bernstein, D., & Ni, L. (2013). *Accelerating K-12 computational thinking using scaffolding, staging, and abstraction*. In Proceeding of the 44th ACM technical
[9] Sanz, A. (2015). Why Teaching and Learning How to Code in Schools. Ed Tech Review.[2020, Jan 6] http://edtechreview.in /trends-insights/insights/1934-why-teaching-and-learning-how to code-in-schools. symposium on Computer science education (pp. 609-614) ACM.
[10] Martín, J. (2016). Por qué debemos enseñar a los niños a programar. [2020, Jan 2] http://futurizable.com/debemos-ensenar-los-ninos-programar.
Ministry of education. (2013). Computer and Information Technology Curriculum Document for the Secondary Stage. Riyadh: Company Development of educational services.

[11] Abu Somra, M. & Attiti, M. (2020). *Scientific Research Methods: from Clarification to Empowerment*. Amman: Al-Yazouri Scientific Press.

Majed bin Maili bin Mohammad Al-Zahrani:
Assistant Prof. of Educational Technologies
Umm Al-Qura University