IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 19, 2020, accepted April 1, 2020, date of publication April 10, 2020, date of current version April 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2987254

Impact of Programming Exposure on the Development of Computational Thinking Capabilities: An Empirical Study

CRISTINA CACHERO[1], **PILAR BARRA**[2], **SANTIAGO MELIÁ**[1], AND **OTONIEL LÓPEZ**[3]

[1]Departamento Lenguajes y Sistemas Informáticos, Universidad de Alicante, 03690 Alicante, Spain
[2]Departamento de Turismo, Universidad Católica de San Antonio Murcia (UCAM), 30107 Murcia, Spain
[3]Departamento de Ingeniería de Computadores, Universidad Miguel Hernández, 03202 Alicante, Spain

Corresponding author: Cristina Cachero (ccachero@dlsi.ua.es)

This work was supported in part by the Spanish Ministry of Economy, Industry, and Competitiveness (Access@City) under Contract TIN2016-78103-C2-2-R, in part by the Spanish Ministry of Science, Innovation, and Universities under Contract RTI2018-098156-B-C54, in part by the Co-Financed by FEDER funds (MINECO/ FEDER/UE), and in part by the EduTech Project co-funded by the Erasmus+ Programme of the European Union under Grant 609785-EPP-1-2019-1-ES-EPPKA2-CBHE-JP.

ABSTRACT Today's digital society has turned the development of students' computational thinking capabilities into a critical factor for their future success. As higher education institutions, we need to take responsibility for this development in every degree course we offer, and provide students with the kind of subjects and activities that best contribute to this aim. In this paper, we study the impact of following an introductory programming course on the development of the computational thinking capabilities of university students. In order to achieve this aim, a concurrent cohort observational study was carried out in which we measured both the subjective and objective computational thinking capabilities of 104 participants (50 first year students enrolled on a Bachelor's degree course in Psychology at the Catholic University of Murcia (UCAM), and 54 first year students enrolled on a Bachelor's degree course in Health Information Systems at the University of Alicante (UA)). The statistical procedures applied to test our hypotheses were a two-way mixed ANOVA, a paired-sample T-test and an independent-sample T-test. The data shows that the group at UA had an initial higher subjective perception of their computational capabilities than the group at UCAM. This perception was supported by their objective scores, which were also significantly higher. However, the subjective assessment of computational capability of the UA group diminished after exposure to the programming course, contrasting with the fact that their objective computational capabilities improved significantly. In the UCAM group, both subjective and objective capabilities remained constant over time. Based on these results, we can conclude that computational thinking capabilities are not developed naturally, but need to be trained. Providing such training to all our students, and not only to those enrolled on undergraduate degrees in engineering, is of paramount importance to allow them to face the challenges of their future professions. This paper empirically demonstrates the extent to which exposing subjects to a programming course may contribute to this aim.

INDEX TERMS Programming, computational thinking, problem-solving, career development, technology social factors, observational study.

I. INTRODUCTION

Our society has already become digital: we live surrounded by programmable objects controlled by software [1]. In this context, students need to develop a set of computational competences that can facilitate their full and effective participation in this new digital reality: the choice is between *programming or being programmed* [2]. These competences are commonly referred to as Computational Thinking (CT) capabilities, and can be formally defined as "the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent" [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

According to this definition, being CT proficient refers to managing a set of problem-solving cognitive processes, as follows [4]:

- Decomposition: Breaking down data, processes, or problems into smaller, more manageable parts;
- Pattern recognition: Observing patterns, trends, and regularities in data;
- Abstraction: Identifying the general principles that generate these patterns;
- Algorithm design: Developing step-by-step instructions for solving these and similar problems.

These processes support problem solving across a myriad of disciplines [5], including maths, science, and humanities [6]. For example, decomposition may help a literature student to break down a poem for analysis, and pattern recognition may help an economist to find cyclic patterns in the rises and falls of a country's economy [4].

Additionally, improving the CT capabilities, regardless of the discipline, has been postulated to be related to noncognitive variables and related soft skills such as tolerance for ambiguity, self-confidence, persistence, creativity and teamwork, among others [4], [7].

A. CT AND PROGRAMMING

Despite the importance of CT in the resolution of various kinds of problems that do not directly involve programming tasks [5], the development of CT processes is often associated with becoming proficient at solving coding activities [8]. For this reason, we have witnessed in recent years the proliferation of block-based programming environments (BBPEs) (e.g., Scratch [9], App Inventor [10] or BitBloq [11]) and platforms (e.g. Code.org [12] or Tynker [13]) whose aim is to facilitate the introduction of people into the programming world as a way to improve their CT capabilities [14].

In order to assess the evolution of students' CT skills due to their engagement in programming activities, a programming-related CT framework has been proposed. This framework has three key dimensions [14]:

- Computational concepts: These are the programming concepts with which designers engage as they program. These include sequences, conditionals, loops, parallelism, events, operators, and data. Computational concepts define the ''what'' of the learning process.
- Computational practices: These are the practices that designers develop as they engage with the above concepts, such as incremental and iterative approaches, testing and debugging, reusing and remixing, and abstracting and modularizing (building something large by putting together collections of smaller parts). Computational practices represent the ''how'' of the learning process.
- Computational perspectives: These are the evolving understandings that designers form of themselves, their relationships to others, and the technological world around them. These involve *expressing* (changes of role from consumer to creator), *connecting* (creating with others and for others) and *questioning* (empowering people to ask questions about and with technology).

We agree with the authors of [8], [14], [15] that engaging in programming is valuable for developing CT capabilities. However, to the best of our knowledge, the research community suffers from a scarcity of empirical data that would allow us to ascertain the real effects of such exposure to programming. This means that important research questions such as ''to what extent does acquiring programming skills help in developing CT capabilities'', ''how much programming exposure must students get in order to increase their CT to a certain level'', or ''to what extent are the CT capabilities improved by the mere exposure to a technology-rich environment, as opposed to specifically engaging in some specific programming training'' remain open.

The aim of this paper is to provide empirical evidence that can help to answer these questions by presenting an observational study of the relationship between improvements in CT capabilities and programming training.

The paper is structured as follows: in Section II, we present the state of the art regarding CT. In Section III, we describe the design of our study, including the research questions, variables and hypotheses. Section IV explains the execution of the study. The data gathered is analyzed in Section V, and the main threats to the validity of the study are examined. Lastly, a discussion of the results and some further lines of research are outlined in Section VI.

II. RELATED WORK

Since the first definition of CT by Papert in 1993 [16], and particularly after the influential paper by Wing [17], there has been a lively discussion about what CT is and how to develop and assess it at all educational levels.

A recent systematic mapping study [18], which focused on the definition, scope and theoretical basis of CT, revealed that, during the period 2006-2014, most research centred on the design of activities to promote CT in the curriculum. According to this study, two of the main strategies for developing CT are game-based learning and constructivism. IN terms of the targeted population, 37.6% of the papers focused on the K-12 level, 24.8% on higher education levels and the remaining 37.6% on both populations. The mapping concluded that the discipline needs to mature and provide (a) agreed-upon theoretical frameworks to sustain the different proposals, and (b) valid assessment instruments that serve to systematically measure the effect of treatments on CT skills.

Similar conclusions appear in [19], in which a review of the state of the art in the K-12 CT field in the years up to 2012 concluded that the research community has already provided broadly agreed-upon CT definitions, and that extensive work has been carried out on the design of activities to develop CT. However, according to the authors, the issue of how to measure CT remains underdeveloped and under-researched.

The issue of CT assessment was specifically addressed in another systematic literature review presented in [20], which concluded that such CT assessment is in its infancy. It also presented an updated picture of the importance of CT for educational institutions, the wayis in which it is being incorporated into already existing subjects/courses in different disciplines, how it is being taught, and a list of available tools. In [21] the authors presented a review of the main CT assessment instruments for middle-school and/or high-school students, together with a classification depending on the evaluative approach used. They proposed the use of a combination of assessment methods or ''system of assessment'' in order to provide a comprehensive evaluation of CT interventions.

It is important to note that CT assessment is viewed as the main weakness of the field in all of the papers reviewed. Although a myriad of proposals have been put for ward [22]–[25], and the interest in the promotion and assessment of CT is high among the research community, there is still a lack of standardised measurement instruments that are open for general use. These instruments also need to be based on agreed-upon theoretical models; otherwise, it is difficult to provide the research community with sound, reliable data that can allow them to provide an objective picture of the true impacts of activities aimed at training the CT capabilities proposed by the educational community in different contexts and at different educational levels.

III. EXPERIMENTAL DESIGN

We conducted an observational study during the period February to June 2018. Observational studies are a kind of empirical study in which, unlike experiments or quasy-experiments, the independent variables are not manipulated but are instead observed, and based on these observations, the researcher tries to draw some conclusions [26]. Hence, in observational studies, the decision regarding who receives an intervention is determined by individual preferences, practice patterns, or policy decisions, rather than being randomised [27].

Our observational study falls under the concurrent/prospective cohort category, in which subjects are followed over time [27]. Cohort studies begin with individuals with and without exposure to a given factor (in our case, to a programming course) who are then evaluated on the subsequent development of an outcome (in our case, their objective and subjective CT scores). The appropriateness of this study design is supported by the fact that (i) there is good evidence to suggest an association between exposure to programming and improvement in CT capabilities; (ii) the interval between exposure to programming training and CT improvement is relatively short, which can minimise loss to follow-up; and (iii) a CT improvement is expected for most subjects, meaning that we can measure CT improvement with a reasonable cohort size- [27].

The main disadvantage of observational studies is that they do not permit us to establish cause-effect relationships, since there is a lack of control over the confounding factors (that is, alternative explanations for the results of the study) [27].

A. OBJECTIVES AND CONTEXT DEFINITION

Following the structure of the Goal-Question-Metric (GQM) template [28], the purpose of this study was to assess the effect of exposure to programming training on both subjective CT auto-perception (SCT) and the objective CT score (OCT) of students enrolled on undergraduate degrees. The SCT score captures what students think about their proficiency with CT, and can be regarded as a measure of self-efficacy, while the OCT score reflects the actual ability of the subjects when carrying out computational tasks.

The population for the study was made up of students who were enrolled on (a) a Bachelor's Degree course in Health Information Systems at the University of Alicante (UA); and (b) a Bachelor's Degree course in Psychology at the Catholic University of Murcia (UCAM), during the second semester of the 2017/18 academic year.

Python was chosen as the programming medium for the treatment group due to its interactive environment, its ability to let novice programmers quickly write non-trivial programs, its adoption by many scientific communities, and its support for numerous specialist libraries [29], [30]. Python can also be executed efficiently, making it a good vehicle not only for small-scale experimentation, but also for larger datasets and longer computational problems [29].

The research questions (RQ) addressed in this study were designed to be answered using quantitative data. The questions were as follows:

- RQ1: Do the students' OCT scores vary depending on their exposure to four months of programming training?
- RQ2: Do the students' SCT scores vary depending on their exposure to four months of programming training?
- RQ3: Do the students' SCT/OCT scores differ between the UA and the UCAM groups?

B. EMPIRICAL STUDY DESIGN

In this study, we planned to gather data from 178 students, 88 of whom were second-semester students on a Health Information Systems degree course at the UA (the treatment group), and 90 of whom were second-semester students on a Psychology degree course at the UCAM (the control group).

1) VARIABLES

To conduct the study, two independent variables (IVs) were defined:

- Time (T): Categorical value, intra-subject, with two possible values: T1 (Feb18) or T2 (May18).
- Group (G): Categorical value, inter-subject, with two possible values: UA or UCAM.

At this point it is important to note that both groups contained subjects who were roughly the same age (the median year of birth was 1999 for both groups), and we could therefore assume that both groups had similar environmental technological influences. The level of previous programming experience in both groups was also roughly similar, despite the dissimilarity of the degree courses; only three subjects out of 88 in the UA group had a previous programming experience, while two of 90 in the UCAM group had previous programming experience.

The set of dependent, or measurable, variables (DVs) was defined as follows:

- OCT: This was scored as the number of CT questions answered correctly [0..28] at the beginning (pre-OCT) and end (post-OCT) of the study.
- SCT: This was a self-reported score of the students' general CT capability [1..10] at the beginning (pre-SCT) and end (post-SCT) of the study.

At the beginning of the study, all subjects also filled in a background questionnaire, giving their age, gender and previous programming experience.

2) HYPOTHESES

Based on the literature review presented in Section II, and the research questions and variables described above, a set of null and alternative hypotheses were defined. For the sake of clarity, next we are only listing the alternative version here, as it is easier to understand:

- $H1_A$: The OCT score changes differently over time depending on whether or not students are exposed to a programming training course.
- $H2_A$: The mean OCT score differs between the UA and the UCAM groups.
- $H3_A$: The SCT score changes differently over time depending on whether or not students are exposed to a programming training course.
- $H4_A$: The mean SCT score differs between the UA and the UCAM groups.

The corresponding null hypotheses (which we aimed to refute via the hypothesis refutation method) simply state that there are no significant OCT/SCT differences between the conditions being compared in each case.

3) OCT AND SCT MEASURING INSTRUMENTS

Given the lack of standardised CT assessment instruments (see Section II) and our need for an instrument that could be administered to 18-year-old students with no previous programming knowledge, we chose the Computational Thinking Test (CTT) [31], as this does not require any programming experience. It has also been thoroughly validated in the Spanish context, and has demonstrated high levels of concurrent (with respect to the PMA, RP30, and FI-R instruments [32]), discriminant, convergent (with respect to Dr. Scratch [33] and Bebras [34]) and factorial validity [35].[1]

This test includes two scales:

- An OCT scale: 28 items, each containing four options, where each item contributes equally to the final score.
- An SCT scale: A single 10-point item.

TABLE 1. Final distribution of subjects by group.

	Enrolled	T1: Feb18	T2: May18	Both
UA	88	65	56	54
UCAM	90	68	66	50
Total	178	133	122	104

IV. EXECUTION OF THE STUDY

The observational study was conducted in two sessions. Each session was held in parallel for the UA and UCAM groups.

The first test session took place during the second week of February 2018. In this session, students were asked to fill in both a background questionnaire and a CT questionnaire. Subjects were not aware in advance that they would be asked to complete these questionnaires, nor did they receive any kind of feedback on their CT performance until the end of the session. Two lecturers supervised each session in order to avoid interactions between subjects. Of the 178 possible students, 135 were present on the day of the study. Since they did not know that they were going to participate in the study, we can assume that the absence of these students had nothing to do with the study, and we therefore consider that their absence did not pose a risk to the validity of the results. For ethical reasons, at the beginning of the session we explicitly asked each subject for permission to treat their data in an anonymised and aggregated way. 133 students (out of 135) accepted.

Between February and May, the treatment group received 60 hours of training in Python. The training was divided into 30 sessions, in which the students received short explanations introducing the main Python concepts, followed by guided practical sessions in which they were asked to solve, on an individual basis, a series of increasingly complex programming problems, many of which were related to their area of expertise. The control group did not receive any kind of programming training.

The second test session took place during the third week of May 2018. Again, subjects did not receive feedback and were supervised by two lecturers at each university. This time, 122 subjects were present. Since these students were not all the same ones that had participated the first time, the final number of subjects included in our study was 104: 54 in the UA group, and 50 in the UCAM group. Table 1 shows the final distribution of subjects by group.

All of the measures were automatically calculated based on the results of the CT questionnaire.

V. DATA ANALYSIS

To analyse the data, we used the SPSS Statistics v.23 software package. Table 2 shows the descriptive statistics corresponding to the measures included in our study.

In Table 2, the colums marked ''UA/UCAM Pre'' refer to the OCT/SCT scores at the beginning of the term, while those marked ''UA/UCAM Post'' refer to the same students' scores

[1] The study questionnaire can be found (in Spanish) at https://ua.eu.qualtrics.com/jfe/form/SV_2f6sdYBk6TwHlNH

TABLE 2. CT measures: Descriptive Statistics.

	UA Pre		UCAM Pre		UA Post		UCAM Post	
	M	*SD*	*M*	*SD*	*M*	*SD*	*M*	*SD*
OCT	19.54	4.31	17.26	4.69	21.20	4.14	17.50	5.06
SCT	6.04	2.07	4.88	2.54	5.59	1.89	4.72	3.09

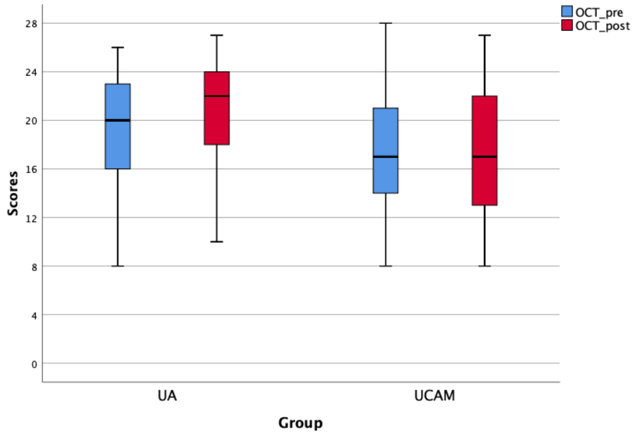

FIGURE 1. Comparison of pre-OCT and post-OCT mean scores of the UA and the UCAM groups.

at the end of the term. The table shows both the students' mean score (M) and the standard deviation (SD).

The statistical procedure initially chosen to test both DVs (OCT and SCT) was a two-way mixed design ANOVA ($\alpha = 0.5$) with an inter-subject factor and an intra-subject factor. This test has eight assumptions that need to be checked before being applied. The first three relate to the study design: we need a continuous DV (in our case we have two: the OCT and SCT), one between-subjects factor (which in our case is the group) and one within-subjects factor (which in our case is time). The other five assumptions relate to how our data fits the two-way mixed ANOVA model, as discussed below.

A. ANALYSIS OF THE OBJECTIVE COMPUTATIONAL THINKING CAPABILITIES: OCT

In order to investigate the effects of a programming course on the development of the students' OCT capabilities ($H1_A$) and the influence of the Group variable on the development of these OCT capabilities ($H2_A$), the first step is to check the assumptions of the two-way mixed ANOVA regarding the pre-OCT and post-OCT DV. These are the absence of outliers, normal distribution of residuals, equal variances between categories, similar covariances, and sphericity, if applicable.

An analysis of both the pre-OCT and post-OCT DVs showed that there were no outliers in the data, as assessed by examination of studentised residuals for values greater than ±3. A visual inspection of their boxplots also showed that there were no values greater than 1.5 box-lengths from the edge of the box (see Fig. 1).

A two-way mixed ANOVA also assumes that the residuals are normally distributed in each cell of the design. This assumption holds for the OCT scores, which were normally distributed for all the groups (z-values for skewness and kurtosis fell in the range ±2.58 for all the cells).

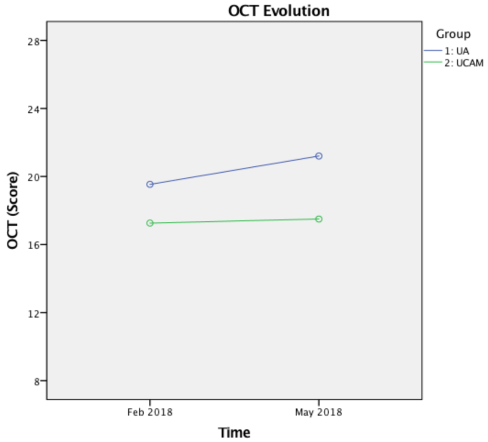

FIGURE 2. Evolution of OCT mean scores (pre-OCT vs post-OCT) along time for both the UA and the UCAM groups.

The third assumption is that there are equal variances between the categories of the between-subjects factor, (Group), and in each category of the within-subjects factor (Time), for the DV (OCT). Homogeneity of variance was shown, as assessed by Levene's test ($p > 0.05$ both for pre-OCT and post-OCT).

A further assumption of the two-way mixed ANOVA is that there are similar covariances. This assumption also holds, as assessed by Box's test of equality of covariance matrices ($p = 0.434$).

Lastly, the assumption of sphericity was not tested, as our within-subjects factor had only two categories.

Since our data fitted the two-way mixed ANOVA model, we applied this test in order to determine whether there was a Group*Time interaction. From Table 2, we can observe how the OCT measure remains practically constant over time for the UCAM group, while there is an improvement of roughly two points for the UA group. This is also shown in Fig. 2, where we can observe that the two lines are not parallel. The results of applying a two-way mixed ANOVA corroborate this perception, and show that the Group*Time interaction is statistically significant: $F(1, 102)$=4.123, $p < 0.05$, partial $\eta^2 = 0.039$.

This result implies that the OCT score changes differently in the UA and UCAM groups, and we can therefore reject hypothesis $H1_0$ which assumes that OCT changes the same way in both groups. The Group*Time interaction qualifies the results of the analysis of each IV and prevents us from analysing the main effects with this test. What we can do, however, is to analyse the simple main effects of the Group and Time variables on OCT independently.

In order to test the simple main effect of the Group variable on the pre-OCT score (taken February 2018), we need to apply an independent-samples T-test. The statistical analysis shows that we can assume equality of variances (Levene's p=0.424). This test also shows that the effect of the Group

TABLE 3. Summary of OCT and SCT testing results.

H	Explanation	Test	Result	p value	Explanation
H1	Impact of the Time Variable on the development of OCT capabilities	Two-way mixed design ANOVA: Group*Time interaction. OCT Scores.	F(1,102)=4.12	p<0.005	The evolution of the OCT score from February to May was significantly different between the UA and the UCAM groups (see Fig. 2).
		Paired samples T-test for UCAM: Simple main effect of the Time variable on the OCT scores of the UCAM group	t(49)=-0.441	p=0.661	Students in the UCAM group, who had not been exposed to any kind of CT training, showed not significant differences in their level of OCT between February and May (see Fig. 2).
		Paired samples T-test for UA: Simple main effect of the Time variable on the OCT scores of the UA group.	t(53)=-3,702	p=0.001	Students in the UA group, who had been exposed to a programming training course, obtained significantly higher OCT scores in May than they did in February (see Fig. 2).
H2	Impact of the Group Variable on the development of OCT capabilities	Two-way mixed design ANOVA: Group*Time interaction. OCT scores.	F(1,102)=4.12	p<0.005	The evolution of the OCT scores from February to May was different between the UA and the UCAM groups.
		Independent samples T-test: Simple main effect of the Group variable on OCT scores in February	t(102)=6.65	p=0.011	In February, the OCT score for students in the UCAM group was significantly lower than for students in the UA group.
		Independent samples T-test: Simple main effect of the Group variable on OCT scores in May	t(102)=4.1	p<0.0005	In May, students at the UCAM group had an OCT score even more significantly lower than students in the UA group.
H3	Impact of the Time Variable on the development of SCT capabilities	Two-way mixed design ANOVA: Group*Time interaction. SCT Scores.	NA		Test assumptions were not met by the data.
		Paired samples T-test UCAM: Simple main effect of the Time variable on the SCT scores of the UCAM group	t(49)=0.629	p=0.53	Students in the UCAM group, who had not been exposed to any kind of CT training, showed not significant differences in their level of SCT between February and May
		Paired Samples T-test UA: Simple main effect of the Time variable on the the SCT scores of the UA group.	t(53)=2.23	p=0.03	Students in the UA group, who had been exposed to a programming training course, had significantly lower SCT scores in May than they had had in February.
H4	Impact of the Group Variable on the development of SCT capabilities	Two-way mixed design ANOVA: Group*Time interaction. SCT Scores.	NA		Test assumptions were not met by the data.
		Independent samples T-test: Simple main effect of the Group variable on SCT scores in February	t(102)=2.555	p=0.012	In February, the SCT score for students in the UCAM group was significantly lower than for students in the UA group
		Independent samples T-test: Simple main effect of the Group variable on SCT scores in May	t(70.301)=2.95	p=0.004	In May, the SCT score for students in the UCAM group was still significantly lower than for students in the UA group

variable is significant: $t(102) = 6.649$ $p = 0.011$. This means that students in the UCAM group (enrolled on a Psychology degree course) scored significantly lower on OCT than students at the UA group (enrolled on a Health Information Systems degree course) when they were measured in February. When they were measured again in May (post-OCT, i.e. after the UA group had enrolled on a programming course), this difference had become even larger (Levene's p=0.088, $t(102) = 4.098, p < 0.0005$).

In order to analyse the simple main effects for the Time variable, since we also have two possible values (February and May), we need to run two separate paired-samples T-tests (one for the UA group and one for the UCAM group). For the UA group, who took a programming course during the semester, the differences were highly significant ($t(53) = -3.702, p = 0.001$). In constrast, for the UCAM group, which was not exposed to any specific CT training, the differences between the February and May scores were not significant ($t(49) = -0.441, p = 0.661$).

A summary of these results can be seen in Table 3.

B. ANALYSIS OF THE SUBJECTIVE AUTO-PERCEPTION ON COMPUTATIONAL THINKING: SCT

In a similar way, in order to test the influence of following a programming course on the students' CT subjective auto-perception (H3$_A$) and the influence of the Group variable on the development of this auto-perception (H4$_A$), the first step is to check whether the pre-SCT and the post-SCT data fit the two-way mixed ANOVA model. Again, there were no outliers in the data, as assessed by examination of studentised residuals for values greater than ±3. However, a visual inspection of the corresponding boxplot showed that three values were greater than 1.5 box-lengths from the edge of the box for the pre-SCT score (see Fig. 3). An examination

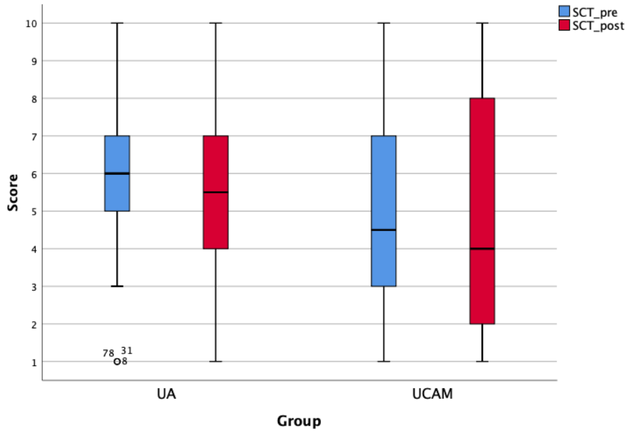

FIGURE 3. Comparison of pre-SCT and post-SCT scores of the UA and the UCAM groups.

of these three points revealed that they were not errors in the data gathering process, but genuinely unusual values, so they were kept in the analysis.

A two-way mixed ANOVA also assumes that the residuals are normally distributed in each cell of the design. In the same way as for the OCT scores, this assumption also holds for the SCT scores, which were normally distributed for all the groups (their z-values for skewness and kurtosis falling into the range ±2.58 for all the cells).

The next assumption is that there are equal variances between the categories of the between-subjects factor (Group), and each category of the within-subjects factor (Time) for the DV, which is the SCT score. This assumption is violated for the post-SCT variable, as assessed by Levene's test of homogeneity of variances ($p < 0.005$). The problem here is that while the skewness for the UA group is slightly negative, it is positive for the UCAM group. We tried both a reflect and square root transformation (good for moderately, negatively skewed data) and a square root transformation (good for moderately, positively skewed data). Neither of these transformations solved the problem; in addition, neither the original values nor any transformations were able to meet the covariance assumption. For this reason, we decided to discard our analysis of the interaction effect and instead to analyze the simple main effects by running two paired samples T-tests for the differences in SCT for each group, and two independent samples T-tests for a comparison of both the pre-SCT and post-SCT measures between the groups.

In order to test the differences between the SCT scores at different points in time with a paired samples T-test for the UA and the UCAM groups, we need one DV measured at the continuous level and one IV that consists of two matched pairs (pre-SCT, measured in February (T1) and post-SCT, measured in May (T2)). There should also be no significant outliers in the differences between the two related groups, and the distribution of the differences in the DV between the two related groups should be approximately normally distributed.

For the UCAM group, no outliers were detected in the data. The differences were normally distributed for both the UA and the UCAM group, with z-values for skewness and kurtosis falling into the range ±2.58 for both groups.

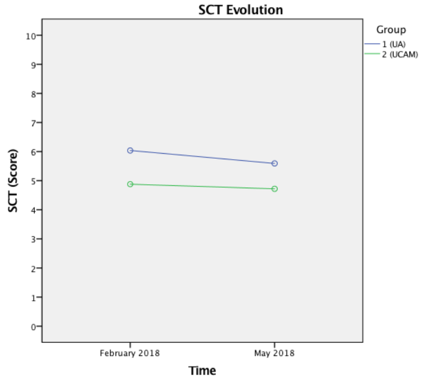

FIGURE 4. Evolution of SCT mean scores (pre-SCT vs post-SCT) along time for both the UA and the UCAM groups.

Lastly, in order to examine the differences in SCT scores between the two groups at both time points (February (T1) and May (T2)) we need to check for outliers and normality. Again, we found nine outliers in the UA group. Inspection of these values showed that there were no errors in the data gathering process, but genuinely unusual values, so they were kept in the analyses.

For the UCAM group, no outliers were detected, with z-values for skewness and kurtosis falling into the range ±2.58 for both groups.

Levene's test for homogeneity of variances showed that although the pre-SCT measure did not violate the assumption, the post-SCT did. A $Log10$ transformation was applied, and although this reduced the problem, it did not prevent the variable from violating the assumption. For this reason, we have not assumed equality of variances for this test.

From Table 2 we can observe how the SCT measure remains practically constant for the UCAM group, while there is a decline for the UA group. This is illustrated in Fig. 4, where we can observe that the two lines show different slopes.

The paired samples T-test for the UA group showed that the SCT scores in February and May differed significantly ($(t(53) = 2,23\ p = 0.03)$. In contrast, this difference is clearly not significant for the UCAM group$(t(49) = 0.629$, $p = 0.53)$.

Finally, regarding the Group variable, the pre-SCT measurement shows significant differences between the UA and the UCAM groups ($t(102) = 2.555, p = 0.012$), with the UA group showing a higher perception of ability. The test for the post-SCT measurement ($Log10$ transformation), which does not assume equal variances, also shows significant differences between both groups ($t(70.301) = 2.946, p = 0.004$), again indicating a higher perception of ability among the UA group.

A summary of these results can be seen in Table 3.

C. THREATS TO THE VALIDITY OF THE STUDY

In our analysis of the main threats to the validity of this study, we use the classification proposed by Cook and

Campbell [36], which is divided into internal, external, construct and conclusion threats.

Threats to internal validity are concerned with the assessment of causality,i.e. with the possibility of hidden factors that may provide alternative explanations for the result. By definition, observational studies have, lower internal validity than experiments, due to the selection bias. This notwithstanding, a cohort longitudinal study is, due to its temporal nature, a design that has higher internal validity than other observational studies. More concretely, our design controls for the effect of the simple passing of time on the objective CT and subjective scores. Students in both groups were unaware that they were being compared with another group, in order to avoid bias in the assessment of the outcome [27]. The initial level of programming experience was also checked, and proved to be similar for both groups. However, we cannot dismiss the possibility that, since they were enrolled on more technical courses, the UA group had been exposed during the period of the study to additional CT stimuli beyond those associated with the programming training, which may account for part of the improvement. The UA and the UCAM group also started from different self-perception levels. A higher initial level of CT self-perception may mean that the subjects are inclined towards a higher level of engagement in CT-related activities, which may also account for part of the improvement. In order to manage this type of threat, replica studies are needed.

Other potential threats to the internal validity are the loss to follow-up and differential loss to follow-up (experimental mortality). Loss to follow-up occurs when individuals drop out during the study period. In our case, 21.8% of the subjects (29 out of 133) dropped out of the study. It is important to note that none of the students that had participated in the first part of the study declined to participate the second time. In addition, students did not know in advance that they were going to be measured twice, nor when these measures would take place, so we can assume that this drop out was not related to the study, and did not affect the results. This risk was unfortunately unavoidable, since for ethical reasons completion of the questionnaires needed to be voluntary, and the workload at the end of the term tends to be high, causing some students to stop attending classes. Differential loss to follow-up is seen when the drop-out rate differs between the exposed group and the group that was not exposed. The drop-out rate for the UA group was 17%, while for the UCAM group (control) it was 27%. Since the higher drop out occurred in the control group (the group that was not receiving any treatment) we can assume that the treatment was not responsible for the drop out rate. The final groups also remained balanced (54 subjects in the treatment group and 50 in the control group), which increases the precision of the study [27].

Threats to external validity are concerned with generalisation of the results. The main threat to external validity here is that the subjects were students of two specific degrees, and the sample was therefore unrepresentative of the overall population of first-year university students. Again, new replica studies are needed in order to mitigate this risk.

Threats to construct validity refer to the relationship between theory and observation. In our study we clearly specified research questions leading to the definition of the study aim and objectives, which in turn led to the CT construct and the way in which it was measured. Our OCT measurement instrument was also thoroughly validated. However, the SCT measure consisted of a single item, and this may have limited its reliability. Unfortunately, to the best of our knowledge there are no other measures for SCT against which we can draw comparisons.

Lastly, *threats to conclusion validity* (also referred to as statistical validity) refer to the relationship between the treatment and the outcome. All the statistical analyses were preceded by tests in order to ensure that the assumptions of the statistical procedure were not being violated. When such assumptions were not met, we applied alternative statistical analyses that were robust to the type of data violation encountered.

VI. CONCLUSION AND FURTHER LINES OF RESEARCH

In this paper we have presented a concurrent cohort observational study that empirically demonstrates the impact that a programming course can have on both OCT and SCT.

Our results suggest that CT capabilities are not developed naturally, but need to be trained. Our data also supports the widespread idea that enrolling on programming courses increases the CT capabilities. In our study, the CT scores for the UA group, who were enrolled on a four-month Python course, increased by 1.66 points (that is, 8.50%) on average. In contrast, the simple passing of time caused an increase of only 1.39% in the CT scores of the UCAM group. Interestingly, despite the increase in their proficiency, the UA group reported a decrease in their technology proficiency self-perception after being exposed to programming challenges, while the self-perception of the UCAM group was maintained. These results may be explained by two widely accepted facts: (i) people tend to over-estimate their skills, and (ii) young people have digital skills gaps that are as wide as in the rest of society [37]. The programming course probably made the UA group aware of their initial over-estimation in February, which may have caused a downward adjustment of their self-perception in May.

These results are in line with the conclusions of the second cycle of the International Computer and Information Literacy Study (ICILS 2018) [38]. This international study, which provides countries with comparable data on students' development of computer and information literacy skills, empirically demonstrated that the development of sophisticated digital skills does not automatically result from growing up with digital devices nor simply from providing students with information and communications technology equipment. Reference [39]. Instead, students need to be taught how to use computers effectively, and teachers need support in their use of CT in teaching. Additionally, the study produced clear evidence of the impact of computer and information literacy (CIL) on student's learning experience, and how CIL and CT skills are required in order to be able to study, work and live in a digital world.

Another important issue arising from the results of the ICILS study establishes the impact of socioeconomic status on digital literacy, with students from higher socioeconomic status backgrounds having significantly higher computer information literacy scores. From our point of view, the existence of this difference supports our claim that university undergraduate degrees should include specific CT training activities in their curricula, so that it is possible to contribute to bridging this gap. However, the issue of how should this be done remains. One possibility might be to design a specific CT course, with or without adaptations, that would be mandatory for all the students, regardless of their degree. One example of such a course is the ''Introduction to Computational Thinking'' proposal from Hambrusch *et al.* [29]. However, there are other possibilities, such as introducing CT-related tasks in a transversal way.

Last but not least, the research community has not yet been able to provide guidance for the process of deciding which course or activity approach would be more effective, both in terms of CT improvement and increasing the student's motivation in different contexts. In view of this, the authors of [29] advocate for the use of a problem-driven approach focused on scientific discovery and computational principles, which includes problems that are directly related to the student's area of interest, although not necessarily within their specialist domain. Although this sounds sensible, we believe that only the provision of standardised measurement instruments and the execution of sufficient empirical studies can shed light on the best approaches for developing CT capabilities with different subject profiles and in different contexts.

Regardless of the option chosen, it seems clear is that we need empirical data to assess which is the best teaching pedagogy for CT. We also need to make sure that the introduction of CT does indeed contribute to improving the student's learning experience.

In order to contribute to filling this research gap, we plan to develop and validate an instrument to measure SCT. We also intend to assess whether the OCT and SCT scores for students enrolled on non-technical courses are affected to the same extent by the exposure to programming as those of subjects enrolled on technical courses. Finally, we are working on a replica study in which students enrolled on other technical courses are measured, in order to check whether the observed behaviour remains consistent. This replica also introduces the execution of two focus groups, one containing CT teachers and another containing a subset of the students enrolled on the study, in order to be able to perform a more in-depth interpretation of the data.

ACKNOWLEDGEMENTS

The authors would like to thank the subjects of the experiment and the local facilitators who kindly agreed to participate in our study.

REFERENCES

[1] L. Manovich, *Software Takes Command*, vol. 5. London, U.K.: A. & C. Black, 2013.

[2] D. Rushkoff, *Program Or Be Programmed: Ten Commands for a Digital Age*. New York, NY, USA: OR Books, 2010.

[3] J. M. Wing. (2010). *Computational Thinking: What and Why?* Accessed: Jan. 20, 2020. [Online]. Available: http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf

[4] Google. *Computational Thinking for Educators*. Accessed: Jan. 20, 2020. [Online]. Available: https://computationalthinkingcourse.withgoogle.com/unit?lesson=8&unit=1

[5] J. M. Wing, ''Computational thinking and thinking about computing,'' *Phil. Trans. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 366, no. 1881, pp. 3717–3725, 2008.

[6] Google. *Exploring Computational Thinking*. Accessed: Jan. 20, 2020. [Online]. Available: https://edu.google.com/resources/programs/exploring-computational-thinking/

[7] M. Román-González, J.-C. Pérez-González, J. Moreno-León, and G. Robles, ''Extending the nomological network of computational thinking with non-cognitive factors,'' *Comput. Hum. Behav.*, vol. 80, pp. 441–459, Mar. 2018.

[8] F. J. García-Pe nalvo, ''What computational thinking is,'' *J. Inf. Technol. Res.*, vol. 9, no. 3, pp. 5–8, 2016.

[9] Scratch. Accessed: Jan. 20, 2020. [Online]. Available: https://scratch.mit.edu/

[10] Scratch. *Mit App Inventor: Anyone Can Build Apps That Impact the World*. Accessed: Jan. 20, 2020. [Online]. Available: http://appinventor.mit.edu/explore/

[11] BQ. *Bitblog: Una Nueva Forma De Programar Facil, Sencilla E Intuitiva*. Accessed: Jan. 20, 2020. [Online]. Available: http://bitbloq.bq.com/

[12] Code.org. *Code.Org*. Accessed: Jan. 20, 2020. [Online]. Available: https://code.org/, last accessed: 20/01/2020.

[13] Tynker. *Tynker: Coding for Kids*. Accessed: Jan. 20, 2020. [Online]. Available: https://www.tynker.com/

[14] K. Brennan and M. Resnick, ''New frameworks for studying and assessing the development of computational thinking,'' in *Proc. Annu. Meeting Amer. Educ. Res. Assoc.*, Vancouver, BC, Canada, 2012, pp. 1–25.

[15] S. Y. Lye and J. H. L. Koh, ''Review on teaching and learning of computational thinking through programming: What is next for K-12?'' *Comput. Hum. Behav.*, vol. 41, pp. 51–61, Dec. 2014.

[16] S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*. New York, NY, USA: Basic Books, 1980.

[17] J. M. Wing, ''Computational thinking,'' *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[18] F. Kalelioglu, Y. Gülbahar, and V. Kukul, ''A framework for computational thinking based on a systematic research review,'' *Baltic J. Modern Comput.*, vol. 4, no. 3, p. 583, 2016.

[19] S. Grover and R. Pea, ''Computational thinking in K–12: A review of the state of the field,'' *Educ. Researcher*, vol. 42, no. 1, pp. 38–43, Jan. 2013.

[20] J. Lockwood and A. Mooney, ''Computational thinking in education: Where does it fit? A systematic literary review,'' 2017, *arXiv:1703.07659*. [Online]. Available: http://arxiv.org/abs/1703.07659

[21] M. Román-González, J. Moreno-León, and G. Robles, ''Combining assessment tools for a comprehensive evaluation of computational thinking interventions,'' in *Computational Thinking Education*. Singapore: Springer, 2019, pp. 79–98.

[22] J. Robertson, ''How to measure computational thinking,'' Heriot-Watt Univ., Edinburgh, U.K., Tech. Rep., 2010. [Online]. Available: https://judyrobertson.typepad.com/judy_robertson/research.html

[23] K. Howland, J. Good, and K. Nicholson, ''Language-based support for computational thinking,'' in *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. (VL/HCC)*, Sep. 2009, pp. 147–150.

[24] S. Brasiel, K. Close, S. Jeong, K. Lawanto, P. Janisiewicz, and T. Martin, ''Measuring computational thinking development with the FUN! Tool,'' in *Emerging Research, Practice, and Policy on Computational Thinking*. Cham, Switzerland: Springer, 2017, pp. 327–347.

[25] L. Werner, J. Denner, S. Campe, and D. C. Kawamoto, ''The fairy performance assessment: Measuring computational thinking in middle school,'' in *Proc. 43rd ACM Tech. Symp. Comput. Sci. Edu. SIGCSE*. New York, NY, USA: ACM, 2012, pp. 215–220.

[26] A. G. Bluman, *Elementary Statistics: A Step by Step Approach*. New York, NY, USA: McGraw-Hill, 2012.

[27] M. D. A. Carlson and R. S. Morrison, ''Study design, precision, and validity in observational studies,'' *J. Palliative Med.*, vol. 12, no. 1, pp. 77–82, Jan. 2009.

[28] D. E. Perry, A. A. Porter, and L. G. Votta, ''Empirical studies of software engineering: A roadmap,'' in *Proc. Conf. Future Softw. Eng.* New York, NY, USA: ACM, 2000, pp. 345–355.

[29] S. Hambrusch, C. Hoffmann, J. T. Korb, M. Haugan, and A. L. Hosking, ''A multidisciplinary approach towards computational thinking for science majors,'' *ACM SIGCSE Bull.*, vol. 41, no. 1, pp. 183–187, Mar. 2009.

[30] E. Freeman, *Head First Learn to Code: A Learner's Guide to Coding and Computational Thinking*. Newton, MA, USA: O'Reilly Media, Inc., 2018.

[31] M. Román-González, ''Computational thinking test: Design guidelines and content validation,'' in *Proc. EDULEARN Conf.*, 2015, pp. 2436–2444.

[32] T. Ediciones. *Tests En línea Tea Ediciones*. Accessed: Jan. 20, 2020. [Online]. Available: http://www.e-teaediciones.com/

[33] G. I. I. del Software Libre, Universidad Rey Juan Carlos. *Dr. Scratch Website*. Accessed: Jan. 20, 2020. [Online]. Available: http://www.drscratch.org/

[34] Bebras. *Bebras: International Challenge on Informatics and Computational Thinking*. Accessed: Jan. 20, 2020. [Online]. Available: https://www.bebras.org/?q=about

[35] M. R. González, ''Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: Validación de un instrumento y evaluación de programas,'' Ph.D. dissertation, UNED, Madrid, Spain, 2016.

[36] T. D. Cook, D. T. Campbell, and A. Day, *Quasi-Experimentation: Design and Analysis Issues for Field Settings*, vol. 351. Boston, MA, USA: Houghton Mifflin, 1979.

[37] *Perception & Reality: Measuring Digital Skills Gaps in Europe, India and Singapore*, ECDL Foundation, Dublin, Republic of Ireland, 2018.

[38] B. Eickelmann, ''Measuring secondary school students' competence in computational thinking in ICILS 2018—Challenges, concepts, and potential implications for school systems around the world,'' in *Computational Thinking Education*. Singapore: Springer, 2019, pp. 53–64.

[39] V. Jacobson. (2019). *ICILS 2018 Results Press Release*. [Online]. Available: https://www.iea.nl/publications/press-release/icils-2018-results-press-release

CRISTINA CACHERO is currently an Associate Professor with the University of Alicante, where she also teaches different courses in the areas of programming and software engineering. Her research topics revolve around the areas of software modeling and empirical software engineering, where she has carried out several evaluations of software engineering techniques, methods, and notations in the context of requirements engineering, model-driven engineering, and user-centered development. She has been awarded several fellowships to support her research work. She has been a Visiting Researcher with the Politecnico de Milano, Italy, with the Gent University, Belgium, with the Université de Montréal, Canada, and with the Universidad de la Frontera, Chile. She is a coauthor of several articles in well-known journals, such as the IEEE Multimedia, the *Journal of Systems and Software* (JSS), the *Journal of Web Engineering* (JWE), *Empirical Software Engineering* (ESE), *Information and Software Technology* (IST), and the *International Journal of Intelligent Systems* (IJIS), and conferences of impact in her research area DEXA, WISE, ER, EC-Web, ICWE, and CAISE. She regularly serves in the PC of conferences and workshops in her area of expertise, and she has also acted as an invited Reviewer in several international journals. She has been a Guest Editor of some special issues in well-known journals, such as JSS or JWE. She has co-organized several workshops in international conferences, such as WTA (SAC 2005), IWWUA (WISE 2008 and WISE 2009), and QWE (ICWE 2010 and ICWE 2011).

PILAR BARRA received the Ph.D. degree in economics from the Catholic University of Murcia, Spain. She is currently working as an Assistant Professor with the Catholic University of Murcia, where she also leads as an Academic Coordinator and the master's degree in innovation and tourism marketing. Her research interests include cultural and educational tourism, economic impact analysis, educational gender differences, and technologies applied to Education. She has published in prestigious journals and has participated as coauthor in book chapters and special reports. She has also participated in several research projects, some of them supported by the European Union.

SANTIAGO MELIÁ received the Ph.D. received from the University of Alicante, in 2007.

He is currently an Associate Professor with the Department of Languages and Information Systems, University of Alicante. His research interests include model-driven development, web engineering methodologies, automatic code generation techniques, and web software architecture. In the last years, he has focused on the empirical software engineering applied to the area of the model-driven for refuting his promises of improvement in productivity, maintainability, and satisfaction in the software development. He has published in prestigious journals, such as the IEEE Internet Computing, the *Journal of Systems and Software*, *Information Systems Frontiers*, the *European Journal of Information Systems*, *Information and Software Technology*, and the *Journal of Web Engineering*, and conferences are OOPSLA, WISE, ER, EC-Web, ICWE, and CADUI. He regularly serves in the PC of several international conferences (WWW, ICWE, and JISBD) and he has co-organized during three years the international workshop MDWE, in 2011, 2012, and 2013. Finally, It is important to highlight that he has coordinated and participated in several industrial research projects in which it has been able to apply the latest techniques of software engineering to develop applications for companies, such as Ambulancias Ayuda S. L. U, INASE, Patronato de Turismo de la diputación de Alicante, Smartloto S. L, and SUMA Gestión Tributaria.

OTONIEL LÓPEZ received the M.S. degree in computer science from the University of Alicante, Spain, in 1996, and the Ph.D. degree in computer science, in 2010.

From 1997 to 2003, he worked as a Programmer Analyst in an important industrial informatics firm. In 2003, he joined to the Computer Engineering Department, Miguel Hernandez University (UMH), Spain, as an Assistant Professor. In 2012, he was promoted to an Associate Professor. He currently leads the GATCOM Research Group (atc.umh.es), Miguel Hernandez University. His research and teaching activities are related to multimedia networking (audio/video coding and network delivery).