



A double dissociative study into the effectiveness of computational thinking

Ana C. Calderon¹  · Deiniol Skillicorn¹ · Andrew Watt¹ · Nick Perham¹

Received: 14 February 2019 / Accepted: 8 August 2019 / Published online: 10 October 2019
© The Author(s) 2019

Abstract

We propose the first steps towards a rigorous analysis of the effectiveness of an emerging pedagogy, Computational Thinking. We found that two aspects of the pedagogy have a positive effect with regard to enhancing two cognitive processes, namely sequential thinking and in abstract thinking. Our data was gathered experimentally with a cohort of mixed-ability undergraduate students enrolled on three distinct courses. The study employed a mixed 2×2 factorial design with type of classroom intervention, measurements were taken at baseline and following delivery of computational thinking methodologies designed to focus on specific components of the pedagogy. The dependent variable was percentage improvement from baseline, and the analyses were conducted using 2×2 mixed ANOVA, an alpha criterion of $p < .05$ was adopted for all analyses. The specific components investigated were algorithmic thinking and abstraction, and we found a positive correlation between enhancements of sequentiality and abstract thinking.

Keywords Computational thinking · Cognitive effects · Pedagogy · Sequentiality · Abstraction

✉ Ana C. Calderon
acalderon@cardiffmet.ac.uk

Deiniol Skillicorn
dskillcom@cardiffmet.ac.uk

Andrew Watt
awatt@cardiffmet.ac.uk

Nick Perham
nperham@cardiffmet.ac.uk

¹ Cardiff Metropolitan University, Llandaff Campus, Cardiff, UK

1 Introduction

Computational Thinking (Papert 1996; Wing 2008) is an increasingly popular pedagogy that teaches problem-solving procedurally by generalising solutions of open-ended problems. It is widely accepted that Computational Thinking (henceforth CT) can be applied to problems in varying disciplines (Guzdial 2008; Wing 2008). However, in pedagogic practice, it is more commonly associated with Computer Science. However, we argue that the wide-range applicability of CT ought not to be taken as an atomic statement, particularly given that, as a pedagogy, its effectiveness remains to be proven. The work presented here proposes, to the best of our knowledge, the first steps towards a rigorous understanding of the cognitive effects of computational thinking on learners.

There are various academic viewpoints of exactly what computational thinking (henceforth CT) entails (NRC 2012). Some researchers follow the original algorithmic definition of (Papert 1980) which defines it in terms of procedural thinking, as the ability to write down algorithms (unambiguous detailed instructions) that can be interpreted and executed by an automated agent. Others view it as a mechanism for empowering learners with the tools to solve complex problems, and focuses on the development of such tools. Currently, there is no consensus regarding the definitive or necessary components of Computational thinking, however, whichever viewpoint is adopted, researchers and practitioners agree it contains at least the following components (Wing 2008): understanding of decomposition, data representation, rule generalization, algorithmic thinking and abstraction. We have focused on the last two CT components because they are the most readily dissociated in terms of the cognitive processes that they must rely upon.

Effective interventions based on CT have the potential to contribute to the development of more effective problem-solvers and will have an impact beyond Higher Education because similar interventions can be applied to primary and secondary education (Grgurina et al. 2015) as well. Our studies are timely aligned with recent policy initiatives that have been designed to improve computing education. The initiatives for such improvement in the U.K. are motivated by its potential economic and societal impact, since it will enable continuing UK leadership in digital innovation (UK Digital Skills Taskforce, 2014; House of Lords 2015). Similarly, the European Commission's science and knowledge service recently published a report providing a comprehensive text on CT skills for pupils in compulsory education, and also some recent findings linking CT to policy-making processes within Europe (Bocconi et al. 2016). We conjectured that two elements of computational thinking, abstraction and sequentiality, would be linked to positive effects on learners ability to, respectively, engage in abstract and sequential thinking. In the following sections we will describe the pedagogical interventions involved in teaching abstraction and sequentiality, the tests we conducted and an analysis of the results. The basis for this statement is founded on the nature of the pedagogical elements, on how

lessons targeting these tools are typically designed, for typical classroom material we refer the reader to (Bell et al. 1998).

2 Pedagogical interventions

We will now describe the pedagogical interventions we conjectured to be responsible for a positive change in sequential thinking and in abstract thinking. Firstly we give more details about computational thinking elements, so that our focused elements (sequentiality and abstraction) can be deservingly placed in context of the pedagogy (CT). As mentioned in the introduction, we take the common view that computational thinking is composed of the following elements:

- **Decomposition**
Understanding a complex problem as the composition of smaller problems. Decomposition is the first step in using computational thinking to solve a complex problem.
- **Data representation**
Data representation includes computational methods for representing information and data. Teaching it in the style of Computational Thinking is done as a generalization of common computer science concepts so they can be applied to varying disciplines. For instance, instead of writing code to represent a problem, students can create visual representations of what the problem or solution are. These are done systematically, and the student is involved in creating the symbolic representations of each concept or solution.
- **Algorithmic thinking and sequentiality**
Similarly to the original specification of computational thinking, algorithmic thinking relates to being able to write solutions to problems by generating ‘algorithms’, i.e. unambiguous step-by-step instructions that can be interpreted and executed by a machine. Our conjecture was that an important part of being able to learn algorithmic thinking involves having a good ability to engage in sequential thinking.
- **Abstraction**
Abstraction is typically the last step involved in solving a problem using computational thinking. After patterns have been recognized, it is generally accepted that students who have mastered abstraction have the ability to correctly identify some detail as “irrelevant” and to ignore it, an important aspect of abstraction is also that it helps learners create a general framework for solving problems in similar categories. This link was unproven prior to this study, and we conducted the first experiment to identify its validity, more details are given in the Results and Method of Analysis sections.

We chose to focus on abstraction and sequentiality as they are the ones we could more readily hypothesise a connection with cognitive effects we could readily investigate.

The investigation was conducted with a cohort of first year students enrolled in two undergraduate courses at Cardiff Metropolitan University, namely Computer Science and Software Engineering. The academic profiles of the cohort made them a good fit to the participant profiles required for our study. This is due to the varied nature of the students. They were learners with mixed ability, but for the vast majority, their performance at secondary school was generally in the mid-percentages. Hence, improvement or lack thereof, with a new intervention, could be comfortably investigated, as it decreases the likelihood of ceiling or floor effects. It would have been more difficult to observe an effect of a new pedagogy if applied to students known to achieve higher (or lower) grades only.

The investigation took place in two sessions of 2 h each, over 2 days in order to avoid fatigue. Students were split into two groups (group A contained 19 students and group B 21 students). The students were enrolled in their first year of their degrees in Computer Science and in Software Engineering. Since they had no previous experience in higher education, it seems fitting to highlight their abilities in secondary school. Of the 19 students in Group A, 2 scored grade B in A-level Computing, 2 scored grade E, 8 scored grade D, and 7 scored grade C. Of the 21 students in group B, 2 scored grade B in A-level Computing, 3 scored grade E, 9 scored grade D, and 7 scored grade C. With regard to the demographics of the groups, of the 19 students in group A, two students associated themselves with the gender female, and the rest male (not necessary the same as their birth gender), the average age was 20.5, minimum age 18 and maximum 22. Of the 21 students in group B, two students associated themselves with the gender female, and the rest male (not necessary the same as their birth gender), the average age was 19, maximum age 20 and minimum 18. The aim in our design was to get an informationally representative sample as oppose to a large statistical sample, and such considerations where included when planning the 2×2 factorial design. For example, participants were purposely sampled to include a range of academic capabilities and experiences to eliminate potential ceiling or floor effects. The generalizability of findings is therefore not lost in such sample sizes, given the study design and choice of analysis.

Both groups participated in both tests, followed by an intervention, and completed by a new battery of tests. Group A was given Intervention 1, which aimed at teaching procedural and sequential thinking to students; and group B intervention 2, aimed at teaching abstraction. The test chosen to measure sequential thinking was an adaptation of the Digit Span (we created two different sequences in the same format as the set provided in the Digit Span subtest, named Set A and Set B). The digit span is a subset of the Wechsler intelligence scale (Wechsler 1949). For this test the participants are shown a sequence of numerical digits and asked to correctly state the sequence, which gets progressively longer with time, we asked for the recall to be done in the normal order. For instance participants are shown a sequence such as 2, 5, 6, 7 and asked to recall it in the correct order; then they are shown a sequence such as 2, 4, 1, 9, 1, 8, 2. The test stops when the participant cannot recall a sequence correctly anymore.

The test chosen for testing abstract thinking was the Tower of London, using Pebl (Mueller and Piper 2014) we choose particular problems and grouped them into two different sets.

Splitting the tests into two sets was necessary as repeating the exact same questions during the tests (before and after interventions) could have lead to skewed results due to participants remembering answers to previous problems.

This is summarized in Table 1 below:

With regard to reliability and validity of the instruments used in the study we highlight that both tests are regularly used for similar tests, see for instance (Hughes et al. 2005; Beaman and Jones 1997; Perham and Vizard 2011; Krikorian et al. 1994; Humes et al. 1997).

Intervention 1 contained an adaptation of Marching Orders Activities (Bell et al. 1998) and an adaptation from the Knight's Tour (Curzon 2014). These activities were adapted by increasing difficulty level, in order to cater for Higher Education, and also to narrow the learning outcomes to focus on procedural and sequentiality, as the original activities focused on nearly the whole range of Computational Thinking components. For the first intervention, students had to take on varying roles for different routine activities and essentially construct an algorithm for solving a particular problem. The first activity consisted of students acting as 'tour guides'; they were asked to figure out routes to take tourists around a city such that all major tourist attractions were visited in the shortest possible length of time. The map drawn was of a fictitious town, and students were expected to draw out the route on individual map copies. They were then told to record their tour, by writing it in a manner as explicit and as unambiguous as possible. Students were asked to complete a similar process with routine activities such as sorting laundry (they had to categorize, then create subcategories, forcing them into algorithmic thinking. This was further enforced by having them recorded their instructions.

Intervention 2 was a mixture of visual abstraction (being able to abstract away from unnecessary detail in geometric images), as well as abstraction in everyday activities. The everyday activities abstraction ensured a similar intervention to that of sequentiality in everyday activities, focusing on different cognitive effects. The visual abstraction exercises consisted of Bongard problems (Linhares 2000; Foundalis 2017).

Table 1 Summary of tests and interventions

	Time 1	Intervention	Time 2
Group A	Digit Span (set A), Tower of London (set A)	Intervention 1 (sequentiality)	Digit Span (set B), Tower of London (set B)
Group B	Digit Span (set A), Tower of London (set A)	Intervention 2 (abstraction)	Digit Span (set A), Tower of London (set B)

These problems are visual abstraction problems and first appeared in the appendix of a book by the mathematician M. Bongard in 1967 (Hofstadter 1980). The problems consist of six boxes on the left and six boxes on the right, and one must find the commonality in the left boxes and in the right boxes, see for instance Figs. 1 and 2. These are notoriously difficult, but the focus was not on pupils' ability to solve them, rather it was on the thought processes involved in attempting to abstract away unnecessary detail from the shapes until a pattern is spotted. They were given a simpler problem, followed by a more complex one, as shown in Fig. 1, more details on how to build such problems, and which ones have been solved so far can be found in (Foundalis 2017). More detail on the interventions is given in the following sections.

The measurements for sequential thinking were taken from one of the Wechsler Adult Intelligence Scale (WAIS) sub-tests, designed to measure cognitive ability in adolescents and adults, namely the digit span subset. We used the Tower of London test (Shallice 1982) (henceforth, TOL) to measure whether abstraction had an impact; the test was conducted with the PEBL test battery (Mueller and Piper 2014). TOL is

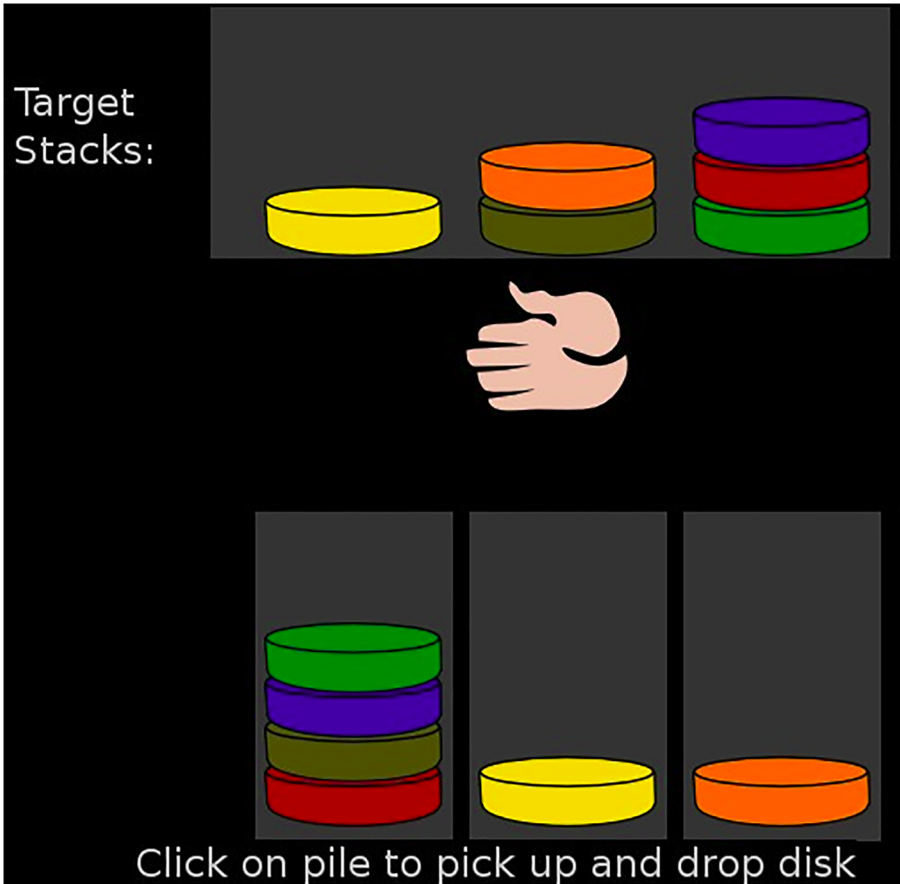


Fig. 1 Screenshot of TOL test, done on Pebl (Mueller and Piper 2014)

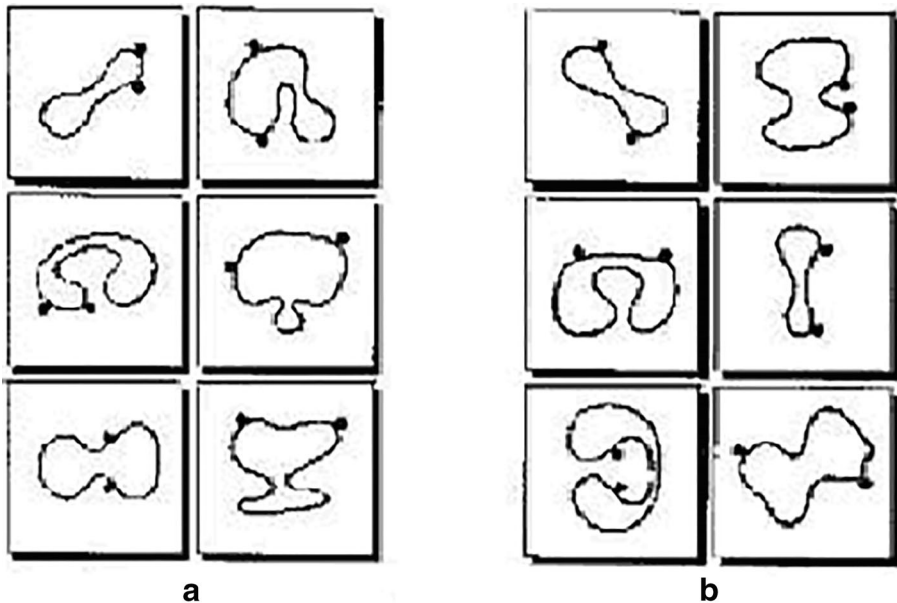


Fig. 2 Bongard example. The solution is that the dots are in the same side of “neck” curvature

typically used to measure deficits in executive functions connected with planning. This is relevant to the initial stages in solving a problem using computational thinking methodology, namely one’s ability to break a complex problem down into smaller pieces, then abstract away irrelevant detail. Essentially, this is the ability to abstractly plan the solution before engaging in solving it. Both interventions are described in detail in the subsequent sections.

3 Intervention 1

The Marching Orders Activities (Bell et al. 1998) essentially teaches algorithmic thinking, in a manner such that a pupil takes on the role of a computer following instructions and another pupil acts as a programmer, giving instructions. For this activity we divided the students in pairs and gave them varying pedagogical activities, modified from the original activities, such as:

- Describing a picture (the student with the role of the computer had to draw it, once the explanation was finished).
- Finding a room in a building on campus without knowing the room number, just by following instructions. The student had to report back the room number to ensure the correct room was found).
- Finding a car in the car park without any details of the car, just by following instructions. The student had to report back on license plate to specify that the right car was found).

An important aspect of this activity is that the programmer had to give the instructions as a whole, and the computer was then expected to follow them correctly. In this instance, ‘correctly’ refers to following the instructions faithfully, even if the instructions lead to a wrong path or diverge from the solution the ‘computer student’ expects.

Students alternated roles for each activity so they could all act as both programmer and computer.

Students were also asked to write down instructions for everyday activities, this forced them to think procedurally about mundane activities, thus reinforcing the notions learned from the Marching Orders activities. Finally, students were asked to complete an activity similar to the Knight’s Tour. Students were placed in groups of 3 or 4, and were given a grid-map, with touristic locations of a fictitious town, and asked to create a solution path for a tourist wishing to visit all points of interest, whilst keeping travel time to a minimum (for instance moving back through an already used path leads to inefficiencies in the solution algorithm).

4 Intervention 2

Students participating in this intervention were asked to describe a cat, then a dog. The rationale behind this activity was that students engaged in abstract thinking by focusing on crucial and necessary aspects of each animal.

They were then asked a series of seemingly simple questions, such as “what did you do on your holiday?”, “what would you like to eat for lunch?” and told to be mindful about how much information was necessary to convey a meaningful answer. Students found this an amusing activity, they were then shown some difficult Bongard problems, with the aim to challenge their ability to visually abstract away from detail (so that the learning was not conditioned to languageability only). They were not expected to solve these problems, the object was to engage their brain in exceedingly complex tasks; solutions were presented after an adequate struggle time was allowed, once students’ attention was turned to things other than the intended problem, the solution was presented, in most cases students re-engaged at this point and spent some time attempting to understand the solution. Students reported that this was a challenging activity.

5 Method of analysis

The objective of the current study was to determine whether two elements of computational thinking (sequentiality and abstraction) have specific effects upon relevant domains of cognitive performance. Digit span and completion time for the Tower of London task were both measured at baseline and following delivery of one of two computational learning classroom interventions, half of the students received a sequentiality intervention whilst the other half received an abstraction intervention.

The study employed a mixed 2×2 factorial design with type of classroom intervention as a between subjects factor with 2 levels (sequentiality vs abstraction) and a within subjects factor with 2 levels (baseline vs post intervention). Analysis was conducted

using a 2×2 mixed ANOVA. Bonferroni adjustment for multiple comparisons was used throughout. An alpha criterion of $p < .05$ was adopted for all analyses.

The dependent variable was percentage improvement from baseline and was calculated separately for the Tower of London and digit span measures because improvement in performance for these 2 measures was in opposing directions (decrease in completion time for the TOL but increase in digit span). Formulae converted the raw test scores into a uniform measure of performance improvement that allowed comparison within a single, conservative analysis. Percentage improvement for the 2 cognitive tests was calculated for each participant using the following 2 formulae:

- 1) Tower of London completion time

$$((\text{baseline} - \text{post}) / \text{baseline}) * 100$$

- 2) Digit span

$$((\text{post} - \text{baseline}) / \text{baseline}) * 100$$

6 Results

Mean percent improvement scores for participants are presented in Fig. 1. Improvement scores for the Tower of London and digit span are presented separately for the participants who received the abstraction and sequentiality interventions. Inspection of Fig. 1 indicates that there were patterns of improvement that were both specific and relevant to the each of the 2 interventions. Specifically, the largest improvement in performance on the Tower of London was in the abstraction group and the largest improvement in digit span was in the sequentiality group. A 2-way interaction in the following analysis would confirm this impression of the descriptive data. There was no main effect of test $F(1,44) = 1.04$, $p > .05$ indicating that the overall size of improvement in the 2 tests was statistically indistinguishable. There was also no main effect of group $F(1,44) < 1$, each of the interventions produced the same amount of overall improvement between baseline and post intervention. Most interestingly there was an interaction between the type of intervention and cognitive test $F(1, 44) = 21.815$, $p < .005$. Analysis of simple effects using pairwise comparisons revealed that improvement in Tower of London completion times were larger for the abstraction group than for the sequentiality group (MEAN DIFFERENCE = 13.37%, $p < .005$), whilst the sequentiality group showed the largest improvement in digit span (MEAN DIFFERENCE = 9.1%, $p < .05$). Unsurprisingly, the improvement in Tower of London completion times was larger than for digit span in the abstraction group (MEAN DIFFERENCE = 8.78%, $p < .05$) and the inverse pattern of difference was

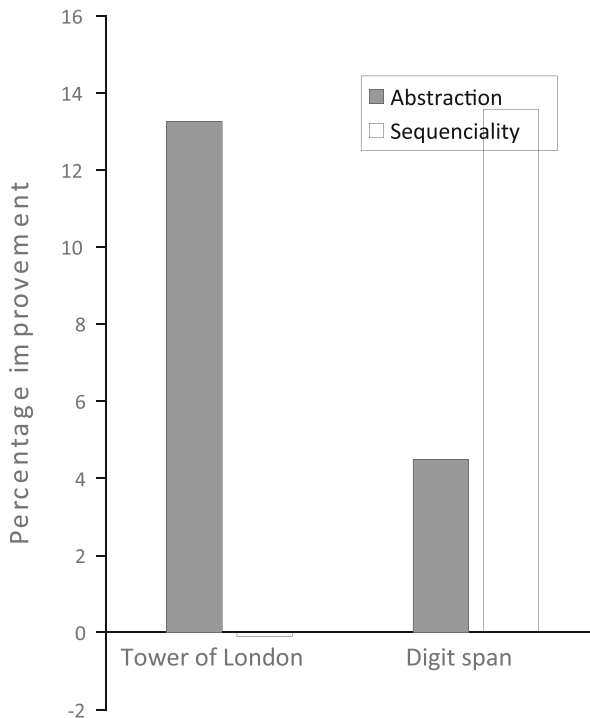


Fig. 3 Mean percentage change in test performance between baseline and post intervention measurements. Error bars show the standard error of the mean

observed in the sequentiality group (MEAN DIFFERENCE = 13.69%, $p < .005$) (Fig. 3).

7 Critique and limitations

The experiment present here is the first step towards providing some context into the efficacy of Computational Thinking as a pedagogy. We have begun with two components that are the most popular in the literature, with regards actionbased reports as there are, presently, no studies on cognitive efficacy (Berland and Lee 2012, NRC 2012, Grover and Pea 2013). Computational Thinking is still a relatively new pedagogy, and the interplay between its different elements is not yet understood, so it could conceivably be the case that the elements investigated are not as important, from a pedagogical stance, as the others. However, as mentioned, most of the work in-classrooms is currently done on these two components, so they a sensible choice for an exploratory study aimed at rooting CT in a firmer context.

8 Conclusion and future work

We have found a positive correlation between teaching abstraction and an enhancement in learner's ability to engage in abstract thinking; and a positive correlation between

teaching algorithmic thinking and an enhancement in sequentiality. With regard to computational thinking, these two components were a sensible place to start because they are the most distant with regard to the cognitive functions they rely on. Now that these positive results have been established we will extend our investigation to other computational thinking components. There is evidence (Zepeda et al. 2015) to suggest that such interventions lead to further motivation in students, hence work in this direction has the potential for positive consequences not only for Computational Thinking, but also for students' engagement with their higher educational journeys.

Acknowledgements Raw data is stored in a repository which will be shared with researchers who request access to any of the authors; provided their intentions are to conduct further analysis, similar investigations, that no financial gain is made from the data, and that the present work is referenced.

This work followed our institution's strict ethical guidelines and was subject to approval by the institution's ethics committee.

The authors have no conflict of interest to report for this work.

The authors are very grateful for the reviewers' comments which have greatly improved the quality of this paper.

Compliance with ethical standards

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed consent Informed consent was obtained from all individual participants included in the study.

Conflict of interest Author Ana Calderon declares that she has no conflict of interest. Author Deiniol Skillicom declares that he has no conflict of interest. Author Andrew Watt declares that he has no conflict of interest. Author Nick Perham declares that he has no conflict of interest.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Beaman, C. P., & Jones, D. M. (1997). Role of serial order in the irrelevant speech effect: Tests of the changing-state hypothesis. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(2), 459.
- Bell, T., Witten, I., and Fellows, M. (1998). Computer science unplugged. www.csunplugged.org. Computer science activities with a sense of fun: Tour guide V1.1 14 Dec 2014) created by Paul Curzon, Queen Mary University of London for Teaching London Computing: <http://teachinglondoncomputing.org>
- Berland, M., & Lee, V. R. (2012). Collaborative strategic board games as a site for distributed computational thinking. *International Journal of Game-Based Learning (IJGBL)*, 1(2), 65–81.
- Bocconi, S., Chiocciariello, A., Dettori, G., Ferrari, A., & Engelhardt, K. (2016). Developing computational thinking in compulsory education-implications for policy and practice (no. JRC104188). Joint Research Centre (Seville site).
- Curzon, P., Dorling, M., Ng, T., Selby, C., & Woollard, J. (2014). Developing computational thinking in the classroom: a framework

- Foundalis, Harry. Research on the Bongard Problems. <http://www.foundalis.com/res/bps/bpidx.htm> Retrieved 2017, May.
- Grgurina, N., Barendsen, E., van Veen, K., Suhre, C., & Zwaneveld, B. (2015, November). Exploring students' computational thinking skills in modeling and simulation projects: A pilot study. In Proceedings of the workshop in primary and secondary computing education on ZZZ (pp. 65–68). ACM.
- Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.
- Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25.
- Hofstadter, D. R. (1979). Gödel, Escher, Bach: an eternal golden braid (Vol. 20). New York: Basic books.
- House of Lords Select Committee on Digital Skills (2015). Make or Break: The UK's Digital Future. <http://www.publications.parliament.uk/pa/ld201415/ldselect/lddigital/111/11102.html> (retrieved: 28 April, 2015).
- Hughes, R. W., Vachon, F., & Jones, D. M. (2005). Auditory attentional capture during serial recall: Violations at encoding of an algorithm-based neural model? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(4), 736.
- Humes, G. E., Welsh, M. C., Retzlaff, P., & Cookson, N. (1997). Towers of Hanoi and London: Reliability and validity of two executive function tasks. *Assessment*, 4(3), 249–257.
- Krikorian, R., Bartok, J., & Gay, N. (1994). Tower of London procedure: A standard method and developmental data. *Journal of Clinical and Experimental Neuropsychology*, 16(6), 840–850.
- Linhares, A. (2000). A glimpse at the metaphysics of Bongard problems. *Artificial Intelligence*, 121(1), 251–270.
- Mueller, S. T., & Piper, B. J. (2014). The Psychology Experiment Building Language (PEBL) and PEBL test battery. *Journal of Neuroscience Methods*, 222, 250–259.
- National Research Council (NRC). (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. Washington, D.C: The National Academies Press. pp. 14–18.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books, Inc.
- Papert, S. (1996). "An exploration in the space of mathematics educations". International Journal of Computers for Mathematical Learning UK Digital Skills Taskforce (2014). Digital Skills for Tomorrow's World. <http://www.ukdigitalskills.com/wp-content/uploads/2014/07/Binder-9-reduced.pdf> (retrieved: 28 April, 2015).
- Perham, N., & Vizard, J. (2011). Can preference for background music mediate the irrelevant sound effect? *Applied Cognitive Psychology*, 25(4), 625–631.
- Shallice, T. (1982). Philosophical transactions of the Royal Society of London. *B*, 298, 199–209.
- UK Digital Skills Taskforce. (2014). Digital skills for tomorrow's world. <http://www.ukdigitalskills.com/> Accessed 15 November 2017.
- Wechsler, D. (1949). Wechsler Intelligence Scale for Children. San Antonio, TX, US: Psychological Corporation
- Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A*, 366(1881), 3717–3725.
- Zepeda, C. D., Richey, J. E., Ronevich, P., & Nokes-Malach, T. J. (2015). Direct instruction of metacognition benefits adolescent science learning, transfer, and motivation: An in vivo study. *Journal of Educational Psychology*, 107(4), 954.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.