

Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: exploring the relationship between computational thinking skills and academic performance

Tenzin Doleck¹ · Paul Bazelais¹ · David John Lemay¹ ·
Anoop Saxena¹ · Ram B. Basnet²

Received: 27 May 2017 / Revised: 10 July 2017 / Accepted: 7 August 2017
© Beijing Normal University 2017

Abstract The continued call for twenty-first century skills renders *computational thinking* a topical subject of study, as it is increasingly recognized as a fundamental competency for the contemporary world. Yet its relationship to academic performance is poorly understood. In this paper, we explore the association between computational thinking and academic performance. We test a structural model—employing a partial least squares approach—to assess the relationship between computational thinking skills and academic performance. Surprisingly, we find no association between computational thinking skills and academic performance (except for a link between cooperativity and academic performance). These results are discussed respecting curricular mandated instruction in higher-order thinking skills and the importance of curricular alignment between instructional objectives and evaluation approaches for successfully teaching and learning twenty-first-century skills.

✉ Tenzin Doleck
tenzin.doleck@mail.mcgill.ca

Paul Bazelais
paul.bazelais@mail.mcgill.ca

David John Lemay
david.lemay@mail.mcgill.ca

Anoop Saxena
anoop.saxena@mail.mcgill.ca

Ram B. Basnet
rbasnet@coloradomesa.edu

¹ McGill University, 3700 McTavish St., Montreal, QC H3A 1Y2, Canada

² Colorado Mesa University, Grand Junction, CO, USA

Keywords Computational thinking · Computational thinking skills · Academic performance · CEGEP students · Curricular alignment

Introduction

Wing's (2006) seminal paper introduced and set out a vision for computational thinking—defined as “taking an approach to solving problems, designing systems and understanding human behavior that draws on concepts fundamental to computing” (Wing 2008, p. 1), later elucidated 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” (Wing 2011, p. 22). This work has motivated a growing stream of research that has both been supportive and, at times, been quite sanguine about computational thinking (Barr and Stephenson 2011; Bundy 2007; Cooper et al. 2010; Gretter and Yadav 2016; Grover and Pea 2013; Guzdial 2008; Lu and Fletcher 2009; Lye and Koh 2014; Snalune 2015; Weintrop et al. 2015; Wing 2006, 2008, 2011, 2014). However, there is general agreement that computational thinking is a fundamental skill that students need to be equipped with.

Computational thinking can best be understood as an umbrella term that relates a subset of related cognitive skills that are involved in computational tasks and activities. Commonly cited examples of computational thinking skills include abstraction, algorithmic thinking, cooperativity, creativity, critical thinking, data analysis, debugging, decomposition, heuristic reasoning, problem solving, and recursive thinking (Barr and Stephenson 2011; Brennan and Resnick 2012; Korkmaz et al. 2017; Wing 2006). However, what is considered computational thinking is still being debated and redefined (Barr and Stephenson 2011; Brennan and Resnick 2012; Gretter and Yadav 2016; Grover and Pea 2013; Korkmaz et al. 2017; Román-González et al. 2017; Sengupta et al. 2013; Voogt et al. 2015; Weintrop et al. 2015). For analytical convenience, we adopt the skills displaying acceptable psychometric qualities identified in the computational thinking scale developed by Korkmaz et al. (2017). Along with Korkmaz et al. (2017), we define computational thinking as being composed of the following skills: algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving. As such, we confine our discussion, and subsequent analysis, of the computational thinking skills to this latter subset.

While computational thinking is not a new notion (Papert 1996), there has been renewed attention for the topic in the educational technology research literature (Lye and Koh 2014). Much of this body of work has been focused on assessing computational thinking (Brennan, and Resnick 2012), conceptualizing the role of and development of programming in computational thinking proficiency (Lu and Fletcher 2009; Lye and Koh 2014), development of students' computational thinking skills (Atmatzidou and Demetriadis 2016), tools designed to foster computational thinking skills (Grover and Pea 2013; Lye and Koh 2014), and integrating computational thinking into the curriculum (Lee et al. 2014; Sengupta et al. 2013), among others.

Despite this increased attention, there is limited empirical research examining the correlation between computational thinking and academic performance. We have limited understanding of how various computational thinking skills are related to and/or influence students' learning and academic outcomes. Given that there have been calls to introduce computational thinking in academia (Voogt et al. 2015; Yadav et al. 2016), there is a clear need for more empirical research to provide more insight into the potential influence of computational thinking skills on learning outcomes. As Wing (2008) asks "how and when should people learn this kind of thinking and how and when should we teach it?" (p. 3720). Such efforts could help provide an informed perspective on the role and implications of computational thinking in students' learning. In addition, a better understanding of how computational thinking skills relates to learning and academic performance could contribute to better curricular alignment by helping to identify curricular objectives that emphasize and promote computational thinking. To that end, the current study provides empirical evidence to further our understanding of the relationship between computational thinking and academic performance.

Literature review: computational thinking competencies

To overcome some of the challenges in advancing computational thinking and integrating computational thinking in academia, Weintrop et al. (2015) suggest that "it will be necessary to break computational thinking down into a set of well-defined and measurable skills, concepts, and/or practices" (p. 130). If we are to evaluate the value and efficacy of embedding computational thinking skills in the curriculum, then valid measures of computational thinking are essential (Román-González et al. 2017). Below, we detail the following computational thinking skills identified by Korkmaz et al. (2017): algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving.

Algorithmic thinking

Computational thinking has been present in the domain of computer science since the 1950s, where it was often phrased as *algorithmic thinking* (Denning 2009). As the field progressed, a distinction evolved between the two terms. Algorithmic thinking stems from the concept of an algorithm, which refers to solving a problem by developing a set of steps taken in a sequence to achieve the desired outcome (Katai 2014). Algorithmic thinking is the thought process towards formulating the steps that leads to the desired result (Hu 2011; Katai 2014), or as stated by Cooper et al. (2010) "algorithmic thinking does not require a computer and mathematical thinking and is almost solely dependent on the human's formalization capacity for abstraction" (p. 28). Concretely, algorithmic thinking is a detail-oriented skill engaging one's cognitive aptitude for comprehending and analyzing problems, developing a sequence of steps towards a suitable solution, streamlining the sequence of steps, and finding substitute steps to ensure that alternate approaches to the solution are catered for (Futschek 2006). Traditionally, computing has followed

an algorithmic structure where an input is received and it is processed sequentially to provide an output; thus, algorithmic thinking is one of the key skills in computational thinking. Indeed, Yadav, Stephenson, and Hong (2017) asserted that “algorithms are central to both computer science and computational thinking. Algorithms underlie the most basic tasks everyone engages in, from following a simple cooking recipe to providing complicated driving directions” (p. 57). Kiss and Akri (2017) found that not having a background in algorithmic thinking handicapped students in higher education and argued that traditional teaching strategies were inappropriate for fostering the conceptual framing required for coding and problem solving. Thus, they highlighted the need for a strategic focus on algorithmic and computational thinking in primary and secondary instruction.

Cooperativity

Social cooperation presents itself as a key approach in computational thinking (Farris and Sengupta 2014; Standl 2016). As the complexity of a problem increases, being able to work collaboratively becomes necessary; students engage higher levels of reasoning as part of computational thinking (National Research Council 2011). According to the National Council for Curriculum and Assessment (NCCA), cooperative problem solving and teamwork are essential for engaging in and learning from program coding specifications (NCCA 2013). Collaborative problem solving as proposed by Warneken, Steinwender, Hamann, and Tomasello (2014) “involves simultaneous coordination of several different behavioural and social-cognitive skills” (p. 49). By working collaboratively, we broaden our thoughts and engage with the thought processes of one or more partner. Standl (2016) addressed collaboration by engaging learners in developing graphics using coding in an environment called ‘Python Turtle.’ The results showed that collaborative problem solving was an effective instructional means that was exhibited in the interactions and communication between the students. Similarly, Farris and Sengupta (2014) studied the development of computational thinking in collaborating students using agent-based modeling. They found that collaboration and having an agent perspective helped in understanding the associated scientific concepts. Looking forward, social cooperation is likely to take on increased importance in computational thinking since new computational problems are increasingly oriented toward large-scale networking and complex data-intensive applications, where solutions result from cooperation and shared problem solving.

Creativity

Creative thinking is a significant aspect of critical thinking and is another dimension of computational thinking (DeSchryver and Yadav 2015). Applying computational thinking principles in problem solving, there is a certain level of creative thinking involved in formulating solutions (Snalune 2015; Voskoglou and Buckley 2012). According to Mishra, Yadav, and the Deep-Play Research Group (2013), computing can be a creative endeavor since it engages cognition and thus unravel creativity, allowing the user to deploy the technology towards creating novel artifacts. Creative

thinking is distinguished from creativity; Sawyer (2012) defines creativity as “a new mental combination that is expressed in the world” (p. 7), while creative thinking is defined by DeSchryver and Yadav (2015) as “cognitive activity comprising various subsets of these component thinking skills that are mediated by the more aesthetic components of traditional creativity” (pp. 413–414). Creative thinking was the idea behind the seminal work of the Lifelong Kindergarten Group (LLK) at the MIT Media Lab that originated the Scratch programming environment. The philosophy behind Scratch, according to Resnick et al. (2009) was to “develop an approach to programming that would appeal to people who hadn’t previously imagined themselves as programmers...make it easy for everyone, of all ages, backgrounds, and interests, to program their own interactive stories, games, animations, and simulations, and share their creations with one another” (p. 60).

Critical thinking

Ater-Kranov, Bryant, Orr, Wallace, and Zhang (2010) highlighted that in addition to problem solving, critical thinking is the other computational skill that is recurrently found in the literature. To engage in problem solving, we need to think at a deeper level and evaluate the problem using or adapting existing knowledge and skill, laying the groundwork for critical thinking. The deeper level of thinking adds a layer of complexity, making critical thinking multidimensional and incorporating skills like evaluation, selection, prediction, abstraction, fostering justified selections, deductions, and generalizations (Kules 2016; Liu and Wang 2010; Williams 2005). The complexity inherent in critical thinking also makes it difficult to get a common definition of the term. Synthesizing from multiple definitions, Voskoglou and Buckley (2012) define critical thinking as “the ability or skill by which the individual transcends his/her subjective self in a wilful manner to arrive rationally at conclusions (not necessarily favourable to him/her) that can be substantiated using valid information” (p. 31). Depending on the complexity of the problem, different levels of thinking, either higher-order or lower-order thinking, are activated. Higher-order thinking is not necessarily algorithmic and produces several solutions since it engages a more cognitively demanding thinking process, while lower-order thinking follows a more straightforward sequential algorithmic style engaging minimal cognitive load, directly arriving to the solution (Mueller et al. 2017; Voskoglou and Buckley 2012). Critical thinking can generate new knowledge since it engages a deeper complex thinking, often resulting in creative solutions, thereby also positioning itself as a precursor to problem solving (Voskoglou and Buckley 2012). Eight thinking attributes have been proposed by the Foundation for Critical Thinking. These attributes conceive of critical thinking in terms of point of view, purpose, question at issue, information, interpretation and inference, concepts, assumptions, and implications and consequences (Foundation for Critical Thinking 2015). These attributes present a synoptic view of critical thinking as a cognitive process: when there is a trigger towards a goal, we begin to question and formulate our understanding based on pre-existing knowledge, and we formulate interpretations and inferences merging our understanding with established concepts, leading to assumptions which entail the implications and consequences that arise from it,

readjusting our point of view (Hu 2011). Critical thinking plays a role in the acquisition of new knowledge as it is only through applying critical thinking to creative thinking and engaging interpretations, concepts, and inferences that new knowledge is created and internalized. Critical thinking promotes skills like creative thinking and problem solving (Voskoglou and Buckley 2012).

Problem solving

Denning (2009) highlights another key aspect of computational thinking as problem solving where an algorithmic solution is pursued for the problem that is structured as information or data (Hu 2011). Google for Education also describes computational thinking as a problem-solving process (Google for Education, n.d.). Polya (1981) initiated research on problem solving and he defined it as “finding a way out of a difficulty, a way around an obstacle, attaining an aim that was not immediately understandable” (p. ix). When we try to find our way out of a problem, we engage ourselves cognitively in the process of finding a solution. Research in the field confirms that problem solving can be considered as the successful outcome of the cognitive engagement process and subconscious thinking towards an obstacle (Voskoglou and Buckley 2012). Peter Henderson (National Research Council 2011) explained CT “as generalized problem solving with constraints” (p. 95) and aptly articulated the relationship of problem solving with computational thinking, elaborating that to achieve a solution, problem solving predominantly engages some form of computation. Barr and Stephenson (2011) further highlighted the need for technology education, suggesting that computational thinking at its core is problem solving that can be executed on a computing device. As most of the research presented here points to the algorithmic nature of problem solving, it appears imperative that students be given the opportunity to apply computational thinking skills to design and implement efficient algorithms in problem solving (Yadav et al. 2017).

Research aims

For the present study, we ask the following question: Is there a relationship between computational thinking skills and academic performance? The overarching goal of this paper is to examine the relationship between computational thinking and academic performance, and specify the extent to which the various components of computational thinking are related to academic performance. To that end, we use a partial least squares (PLS) approach to empirically evaluate the structural relations between computational thinking skills and academic performance. We test a structural model that relates computational thinking skills with academic performance, controlling for age, gender, and prior academic achievement. The research model is presented in Fig. 1.

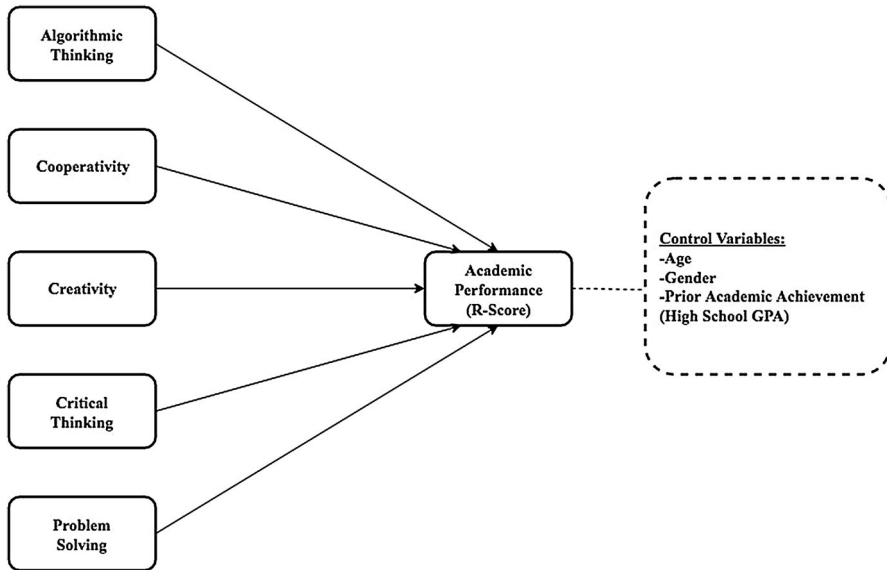


Fig. 1 Research model

Method

Participants, procedure, and measures

Data were collected from pre-university science students completing the DEC (diplôme d'études collégiales) at an English *Collège d'enseignement général et professionnel* (CEGEP; see, for a description, Bazalais et al. 2016) located in northeastern Canada. Usable data from 104 students were part of the final analyses. The convenience sample for the current study was composed of 54 females and 50 males. Participants' mean age was 17.9 years ($SD = 0.74$).

Respondents participated voluntarily and completed a questionnaire composed of two sections. "Introduction" section included demographic and academic information (GPA), and "[Literature review: computational thinking competencies](#)" section contained items related to computational thinking dimensions that were measured using the computational thinking scale (Korkmaz et al. 2017). The computational thinking scale comprises 29 items and is divided into five dimensions: algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving. Items were scored on a 5-point Likert scale ranging from 1 = *Never* to 5 = *Always*. Academic performance was measured using students' self-reported grade point average (CEGEP R-Score). Demographic and prior achievement (age, gender, high school GPA) were employed as control variables in the model; the importance of these control variables have been well documented in the education literature.

Analysis and results

A partial least squares (PLS; Wold 1982) approach was employed because of the exploratory nature of the present study. All data analyses were conducted using the WarpPLS software (Kock 2015a). The recommended two-stage data analysis (measurement model and structural model) approach was followed (Hair et al. 2011; Kock 2015b). In the subsections below, we report the results for both the measurement and structural models. The measurement model offers statistics to establish the reliability and validity of the data and model. The structural model determines the strength and significance of the model relationships.

Measurement model

The model fit was assessed using various indexes; the model fit statistics (Table 1) accorded with the suggested criteria (Kock 2015b) demonstrating a good fit between the model and data. The assessment of the measurement model involved the evaluation of the adequacy of reliability and validity of the constructs in the model, which were tested via individual indicators such as reliability, internal consistency, convergent validity, and discriminant validity.

The indicator loadings—values in the range 0.5–0.7 generally deemed adequate (Hair et al. 2011; Kock 2015b)—were inspected, and items with loadings less than the 0.6 threshold were dropped (Table 2). The internal consistency reliability was assessed using the composite reliabilities (preferred to Cronbach’s alpha). Composite reliabilities, which ranged from 0.832 to 0.910, were above the 0.7 threshold (Hair et al. 2011). All Cronbach’s alpha values, which ranged from 0.671 to 0.867, were near or above the 0.7 threshold (Hair et al. 2011). Convergent validity was assessed using the average variance extracted (AVE). AVEs, which ranged from 0.570 to 0.752, were greater than the 0.5 threshold (Hair et al. 2011). The relevant statistics for composite reliability, Cronbach’s alpha, and AVEs are presented in Table 3.

The constructs are abbreviated to ease readability: algorithmic thinking (ALG), cooperativity (COO), creativity (CRE), critical thinking (CRI), and problem solving (PRO).

Discriminant validity, which demonstrates the degree to which a construct is different from other constructs, was assessed using the Fornell–Larcker criterion

Table 1 Model fit statistics

Measure	Values	Recommended criterion
Average path coefficient (APC)	0.154, $P = 0.026$	Acceptable if $P < 0.05$
Average R-squared (ARS)	0.470, $P < 0.001$	Acceptable if $P < 0.05$
Average adjusted R-squared (AARS)	0.425, $P < 0.001$	Acceptable if $P < 0.05$
Average block VIF (AVIF)	1.270	Acceptable if ≤ 5
Average full collinearity VIF (AFVIF)	1.621	Acceptable if ≤ 5

Table 2 Loadings and cross-loadings of measurement items

	ALG	COO	CRE	CRI	PRO	P value
ALG2	0.847	-0.147	-0.208	-0.040	0.201	<0.001
ALG3	0.770	-0.057	-0.012	-0.207	0.179	<0.001
ALG4	0.777	0.021	0.042	0.086	-0.208	<0.001
ALG5	0.771	0.121	0.075	-0.054	-0.016	<0.001
ALG6	0.704	0.084	0.134	0.238	-0.191	<0.001
COO1	-0.069	0.829	-0.122	0.215	-0.080	<0.001
COO2	-0.107	0.862	0.099	-0.081	0.006	<0.001
COO3	0.011	0.893	-0.110	0.121	-0.015	<0.001
COO4	0.175	0.799	0.142	-0.271	0.093	<0.001
CRE4	0.053	-0.046	0.849	0.139	-0.067	<0.001
CRE5	0.118	0.092	0.613	-0.177	0.140	<0.001
CRE8	-0.132	-0.019	0.889	-0.011	-0.032	<0.001
CRI1	-0.050	0.138	-0.077	0.718	0.092	<0.001
CRI2	0.125	-0.049	-0.056	0.794	0.091	<0.001
CRI3	-0.211	0.025	-0.115	0.820	0.044	<0.001
CRI4	0.016	-0.075	0.299	0.702	-0.108	<0.001
CRI5	0.134	-0.039	-0.022	0.734	-0.134	<0.001
PRO2	-0.073	0.081	0.128	0.008	0.867	<0.001
PRO3	0.073	-0.081	-0.128	-0.008	0.867	<0.001

Bold values are indicator loadings

Table 3 Measurement scale characteristics

Construct	Composite reliability (CR)	Cronbach's alpha	Average variance extracted (AVE)
ALG	0.882	0.833	0.601
COO	0.910	0.867	0.717
CRE	0.832	0.693	0.629
CRI	0.868	0.810	0.570
PRO	0.859	0.671	0.752

Table 4 Discriminant validity check

	ALG	COO	CRE	CRI	PRO
ALG	0.775	0.179	0.541	0.639	-0.393
COO	0.179	0.847	0.046	0.120	0.151
CRE	0.541	0.046	0.793	0.599	-0.204
CRI	0.639	0.120	0.599	0.755	-0.314
PRO	-0.393	0.151	-0.204	-0.314	0.867

Bold values represent the square roots of AVEs

(Fornell and Larcker 1981; Kock 2015b). From Table 4, it can be observed that the Fornell–Larcker criterion is met as all the diagonal values (representing the square roots of AVEs) are greater than the off-diagonal numbers (representing the correlations between the variables) in the corresponding rows and columns.

Thus, the acceptability of the psychometric properties of the measurement model was established.

Structural model

Having established the adequacy of the measurement model, the structural model was evaluated to test the relationship between the constructs. Multicollinearity was not an issue since the variance inflation factors (VIFs) between the constructs were below the suggested threshold of 5 (Kock 2015b). Predictive validity (Q^2) was used to assess the predictive relevance associated with each endogenous variable in the model; all Q^2 coefficient values were greater than zero demonstrating an acceptable level of predictive relevance (Kock 2015b).

The structural model was examined to test the hypotheses by examining the path coefficients (β) and path significance (P value). The structural model illustrates the path coefficients calculated for each link in the model, as well as their corresponding P values. Examining the path coefficients, Cooperativity was significantly negatively related to GPA ($\beta = -0.189$; $P < 0.05$). No further support was found for associations between the remaining constructs and academic performance: algorithmic thinking and academic performance ($\beta = 0.124$, $P > 0.05$), creativity and academic performance ($\beta = 0.129$, $P > 0.05$), critical thinking and academic performance ($\beta = -0.002$, $P > 0.05$), problem solving and academic performance ($\beta = -0.123$, $P > 0.05$).

Thus, we find that cooperativity was significantly and negatively associated with academic performance, controlling for age, gender, and prior academic achievement. We find no support for association for algorithmic thinking, critical thinking, problem solving, creativity, and academic performance.

Discussion

Our findings suggest a lack of association between computational thinking skills and academic performance (except for a link between cooperativity and academic performance). This is noteworthy given the importance that has been placed on teaching and learning twenty-first-century skills in various curricular reforms implemented since the turn of the millennium. If there is no relationship between computational thinking skills and academic performance, we must ask whether these curriculum-mandated skills are being explicitly taught at all. More distressingly, we must wonder at measures of academic performance that are negatively associated with cooperativity. This issue is directly related to the problem of curricular alignment (Anderson 2002; Biggs 1996, 1999) or curriculum coherence (Bateman et al. 2008). Briefly, our choices of instructional approaches and evaluation means can have important impacts on the enactment of curricular imperatives.

At the CEGEP level, classroom researchers (Bateman et al. 2008) have explored the effects of curricular alignment in professional development activities on student success. According to Bateman et al. (2008), “A valid curriculum is coherent. Curriculum coherence is the degree to which the intended learning outcomes

(instructional objectives), instructional processes (teaching and learning activities) and assessments (formative and summative evaluations of student learning) are aligned or connected” (p. 22). This professional development effort represented one of the first attempts at a CEGEP-wide curriculum alignment effort.

Instructional objectives may not be attained if they are not properly evaluated (Biggs 1999). In their multi-year action research project, the researchers found that that there was very often a disconnect between espoused course objectives and evaluation tasks, namely that evaluations were not adequately addressing the achievement of the specified learning outcomes. The main outcome of the project was the establishment of a curriculum review process, by which teachers became accountable to their students and to each other. The curriculum review process is “a process that can be used to achieve alignment, equity, fairness, and an increase in learning for...students with a corresponding increase in job satisfaction for...teachers” (Bateman et al. 2008, p. 22).

Curriculum alignment may be difficult to achieve in practice as many competing institutional, political, and social factors intercede on instructional decisions. Moreover, persistent institutional realities have led to situations where teachers often operate in isolation. As Bateman et al. (2008) pointed out, instituting the curriculum review process required a change or paradigm shift in departmental operations. Indeed, the curriculum review process itself became a vehicle for transformational change (Mezirow 2000) and for the development of a community of practice (Wenger 1998) among the CEGEP teachers.

As the calls for twenty-first-century skills go unabated, it appears necessary for educational stakeholders to ensure that there is an alignment between espoused learning outcomes, instructional approaches, and evaluation means. If students are going to develop the computational thinking skills required to succeed today, these skills must be explicitly addressed in a coherently organized and delivered curriculum.

The study has some limitations that must be considered. These limitations also give rise to a host of interesting questions. Demographic limitations include the fact that the convenience sample for the study was drawn from a single pre-university college; thus, generalizability of our findings is a natural concern. Accordingly, similar investigations ought to be expanded to other contexts (such as school levels and student majors) to observe the variability, if any, of the examined associations. We only considered age, gender, and high school GPA as control variables; future studies may consider the incorporation of other salient control variables. Moreover, as a cross-sectional study, the present findings do not permit causal inferences. Some experimental research has been conducted that has explored some causal connections between teaching CT skills and academic performance (Lockwood and Mooney 2017), but this area of research remains in its infancy. More research is needed to document the nature of the relationship between CT skills and academic performance. Future research ought to examine the temporal development of computational thinking to understand if these constructs are static or develop over time. In this study, we did not control for the effects of students’ programming experience, and thus findings could have been potentially affected. Further evidence of validity and reliability of the recently proposed computational thinking scale (Korkmaz et al. 2017) would strengthen our findings. We restricted our

investigation of computational thinking to the skills identified in the computational thinking scale. This constrained scope leaves open the possibility that other salient computational thinking skills were omitted from the present study. An important avenue for future work will be to examine if interventions geared toward developing the identified computational thinking skills oriented any effects on students' learning outcomes.

Conclusion

The present study sought to empirically investigate the relationship between computational thinking skills and academic performance. We found that cooperativity was negatively associated with academic performance, controlling for age, gender, and prior academic achievement. In contrast, we did not find any significant association between the other computational thinking skills and academic performance. Our findings contribute to the computational thinking literature by documenting the relationship between computational thinking dimensions and academic performance. The lack of association between computational thinking and academic performance suggests that better curricular alignment may be necessary if students are to develop computational thinking and other twenty-first-century skills.

References

- Anderson, L. (2002). Curricular alignment: A re-examination. *Theory into Practice*, 41(4), 255–264.
- Ater-Kranov, A., Bryant, R., Orr, G., Wallace, S., & Zhang, M. (2010). Developing a community definition and teaching modules for computational thinking: Accomplishments and challenges. In *Proceedings of the 2010 ACM conference on information technology education* (pp. 143–148). ACM.
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics And Autonomous Systems*, 75, 661–670. doi:10.1016/j.robot.2015.10.008.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12. *ACM Inroads*, 2(1), 48. doi:10.1145/1929887.1929905.
- Bateman, D., Taylor, S., Janik, E., & Logan, A. (2008). *Curriculum coherence and student success*. Champlain College CEGEP. Retrieved from: http://www.cdc.qc.ca/parea/786950_bateman_curriculums_champlain_st_lambert_PAREA_2007.pdf.
- Bazelais, P., Lemay, D. J., & Doleck, T. (2016). How does grit impact college students' academic achievement in science? *European Journal of Science and Mathematics Education*, 4(1), 33–43.
- Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education*, 32, 1–18.
- Biggs, J. (1999). *Teaching for quality learning at university*. Society for Research into Higher Education/Open University Press.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. Vancouver: Paper presented at the American Educational Research Association.
- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67–69.
- Cooper, S., Pérez, L., & Rainey, D. (2010). K-12 computational learning. *Communications of the ACM*, 53(11), 27. doi:10.1145/1839676.1839686.
- Denning, P. (2009). The profession of IT beyond computational thinking. *Communications of the ACM*, 52(6), 28. doi:10.1145/1516046.1516054.

- Deschryver, M. D., & Yadav, A. (2015). Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated approaches to teaching and learning. *Journal of Technology and Teacher Education*, 23(3), 411–431.
- Farris, A. V., & Sengupta, P. (2014). Perspectival computational thinking for learning physics: A case study of collaborative agent-based modeling. In *Proceedings of the 12th international conference of the learning sciences (ICLS 2014)* (pp. 1102–1107).
- Fornell, C., & Larcker, D. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research*, 18(1), 39–50.
- Foundation for Critical Thinking. (2015). Elements and standards learning tool. Retrieved from: <http://www.criticalthinking.org/pages/analyzing-and-assessing-thinking-/783>.
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. In *International conference on informatics in secondary schools-evolution and perspectives* (pp. 159–168). Berlin: Springer.
- Google for Education. (n.d.). CT overview. Retrieved from <https://edu.google.com/resources/programs/exploring-computational-thinking/#!ct-overview>.
- Gretter, S., & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, 60(5), 510–516. doi:10.1007/s11528-016-0098-4.
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. doi:10.3102/0013189x12463051.
- Guzdial, M. (2008). Education paving the way for computational thinking. *Communications of the ACM*, 51(8), 25. doi:10.1145/1378704.1378713.
- Hair, J., Ringle, C., & Sarstedt, M. (2011). PLS-SEM: Indeed a silver bullet. *The Journal of Marketing Theory and Practice*, 19(2), 139–152. doi:10.2753/mtp1069-6679190202.
- Hu, C. (2011). Computational thinking: What it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on innovation and technology in computer science education* (pp. 223–227). ACM.
- Katai, Z. (2014). The challenge of promoting algorithmic thinking of both sciences- and humanities-oriented learners. *Journal of Computer Assisted Learning*, 31(4), 287–299. doi:10.1111/jcal.12070.
- Kiss, G., & Arki, Z. (2017). The influence of game-based programming education on the algorithmic thinking. *Procedia - Social and Behavioral Sciences*, 237(21), 613–617.
- Kock, N. (2015a). WarpPLS. Retrieved from <http://www.warppls.com>.
- Kock, N. (2015b). *WarpPLS 5.0 user manual*. ScripWarp Systems. Retrieved from http://cits.tamui.edu/WarpPLS/UserManual_v_5_0.pdf.
- Korkmaz, Ö., Çakir, R., & Özden, M. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*. doi:10.1016/j.chb.2017.01.005.
- Kules, B. (2016). Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. In *Proceedings of the association for information science and technology*. Silver Springs, MD: American Society for Information Science.
- Lee, I., Martin, F., & Apone, K. (2014). Integrating computational thinking across the K-8 curriculum. *ACM Inroads*, 5(4), 64–71. doi:10.1145/2684721.2684736.
- Liu, J., & Wang, L. (2010). Computational thinking in discrete mathematics. In *IEEE 2nd international workshop on education technology and computer science* (pp. 413–416).
- Lockwood, J., & Mooney, A. (2017). Computational thinking in education: Where does it fit? *A Systematic Literary Review*. (under review).
- Lu, J., & Fletcher, G. (2009). Thinking about computational thinking. *ACM SIGCSE Bulletin*, 41(1), 260–264. doi:10.1145/1539024.1508959.
- Lye, S., & Koh, J. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. doi:10.1016/j.chb.2014.09.012.
- Mezirow, J. (2000). *Learning as transformation: Critical perspectives on a theory in progress*. San Francisco: Jossey-Bass.
- Mishra, P., Yadav, A., & Deep-Play Research Group. (2013). Rethinking technology & creativity in the 21st century. *TechTrends*, 57(3), 10–14.
- Mueller, J., Beckett, D., Hennessey, E., & Shodiev, H. (2017). Assessing computational thinking across the curriculum. In *Emerging research, practice, and policy on computational thinking* (pp. 251–267). Springer International Publishing.

- National Council for Curriculum and Assessment. (2013). Draft specification for junior cycle short course. Retrieved from http://www.juniorycycle.ie/NCCA_JuniorCycle/media/NCCA/Documents/Consultation/Short%20Courses/SC_P_and_C.pdf.
- National Research Council. (2011). Report of a workshop of pedagogical aspects of computational thinking. Retrieved from http://www.nap.edu/catalog.php?record_id=13170.
- Papert, S. (1996). An exploration in the space of mathematics educations. *International Journal of Computers for Mathematical Learning*, 1(1), 95–123. doi:10.1007/bf00191473.
- Polya, G. (1981). *Mathematical discovery: On understanding, learning and teaching problem solving*. New York: Wiley.
- Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., et al. (2009). Scratch. *Communications of the ACM*, 52(11), 60. doi:10.1145/1592761.1592779.
- Román-González, M., Pérez-González, J., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the computational thinking test. *Computers in Human Behavior*, 72, 678–691. doi:10.1016/j.chb.2016.08.047.
- Sawyer, K. (2012). *Explaining creativity: The science of human innovation* (2nd ed.). New York: Oxford Univ. Press.
- Sengupta, P., Kinnebrew, J., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education And Information Technologies*, 18(2), 351–380. doi:10.1007/s10639-012-9240-x.
- Snalun, P. (2015). The benefits of computational thinking. *ITNOW*, 57(4), 58–59. doi:10.1093/itnow/bwv111.
- Standl, B. (2016). A case study on cooperative problem solving processes in small 9th grade student groups. *IEEE global engineering education conference (EDUCON)*, Abu Dhabi (pp. 961–967).
- Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728. doi:10.1007/s10639-015-9412-6.
- Voskoglou, M. G., & Buckley, S. (2012). Problem solving and computers in a learning environment. *Egyptian Computer Science Journal*, 36(4), 28–46.
- Warneken, F., Steinwender, J., Hamann, K., & Tomasello, M. (2014). Young children's planning in a collaborative problem-solving task. *Cognitive Development*, 31, 48–58. doi:10.1016/j.cogdev.2014.02.003.
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2015). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1), 127–147. doi:10.1007/s10956-015-9581-5.
- Wenger, E. (1998). *Communities of practice: Learning, meaning, and identity*. Cambridge: Cambridge University Press.
- Williams, R. L. (2005). Targeting critical thinking within teacher education: The potential impact on Society. *The Teacher Educator*, 40(3), 163–187.
- Wing, J. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33. doi:10.1145/1118178.1118215.
- Wing, J. (2008). Computational thinking and thinking about computing. *Philosophical Transactions Of The Royal Society A: Mathematical, Physical And Engineering Sciences*, 366(1881), 3717–3725. doi:10.1098/rsta.2008.0118.
- Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Newsletter*, 6, 1–32. Retrieved from http://link.cs.cmu.edu/files/11-399_The_Link_Newsletter-3.pdf.
- Wing, J. (2014). Computational thinking benefits society. *Social issues in computing*. Retrieved from <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>.
- Wold, H. (1982). Soft modeling: The basic design and some extensions. In K. Joreskog & H. Wold (Eds.), *Systems under indirect observation* (pp. 1–54). Amsterdam: North-Holland.
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K-12 classrooms. *TechTrends*, 60(6), 565–568. doi:10.1007/s11528-016-0087-7.
- Yadav, A., Stephenson, C., & Hong, H. (2017). Computational thinking for teacher education. *Communications of the ACM*, 60(4), 55–62. doi:10.1145/2994591.

Tenzin Doleck is a doctoral student at McGill University in Montreal, QC.

Paul Bazalais is a doctoral student at McGill University and an instructor at John Abbott College in Montreal, QC.

David John Lemay is a research associate with the Centre for Medical Education at McGill University. He received his PhD in the Learning Sciences in spring 2017.

Anoop Saxena is a doctoral student at McGill University in Montreal, QC.

Ram B. Basnet is an Assistant Professor in the Department of Computer Science at Colorado Mesa University.