Exploring the Relationship Between Coding, Computational Thinking and Problem Solving in Early Elementary School Students

Abstract

As more young children learn to code, it is vital understand how coding influences early development. In this study we explore the relationship between learning to code and unplugged (non-coding) problem-solving skills. Children ages seven to nine participated in a six-week coding curriculum (CAL-KIBO) utilizing the KIBO robot. Participants received assessments of coding proficiency, computational thinking and problem-solving skills. Unplugged problem-solving skills improved over the course of the curriculum and were significantly correlated with end-of-study coding proficiency. Results indicate that learning to code can improve problem-solving skills, particularly in children who generalize the knowledge gained from coding into computational thinking skills. Implications for future elementary coding initiatives are discussed.

Objective

The present analysis is part of a larger study of the CAL-KIBO (Coding as Literacy-KIBO robot) curriculum carried out in Norfolk, VA with support from the Department of Defense Education Activity (DoDEA). The larger CAL-KIBO study examines the impact of the curriculum on coding and literacy skills. The objective of the present analysis is to determine how the acquisition of coding skills through the CAL-KIBO curriculum relates to CT and unplugged (non-coding) problem-solving skills.

Theoretical Framework

In 2006, Jeannette Wing proposed that acquiring computational thinking (CT) skills can have benefits for the development of other domains of thinking. CT includes thought processes such as thinking recursively, applying abstraction when figuring out a complex task and using heuristic reasoning to discover a solution (Wing, 2006; Wing, 2011). Mastery of CT includes the processes of pattern recognition, conceptualization, planning and problem solving. CT skills are not only valuable for computer programming but helpful in a variety of other contexts (Román-González, et al., 2019; Zhang & Nouri, 2019). There is evidence that learning to code can improve the acquisition of CT skills (Grover & Pea, 2013; Lye & Koh 2014; Fraillon et al., 2018). There has been less work done to address whether learning coding and CT skills lead to improved problem-solving abilities in young children. Addressing this gap in knowledge has broader implications for computer science policy and curriculum development.

Multiple studies conducted with university students have shown that baseline problem solving skills can predict later mastery of coding (Barlow-Jones, & Van der Westhuizen, 2017; Lishinski et al., 2016; Nowaczyk, 1984). CT skills measured in middle school students prior to their learning to code predicted their subsequent computational talent (Román-González et al., 2018). This study showed that there was a relationship between coding and CT. In other words, higher baseline CT skills predicts a higher likelihood of success in learning to code. Teaching problem solving before coding has been shown to improve programming proficiency in university

students (Koulouri et al., 2015). Although there has been research in middle school children and adults, there is a paucity of information about the relationship between problem solving abilities and coding in younger children (ages 5-9).

Most prior studies of CT skills in young children employed protocols that required participants to have some proficiency in coding (Bers, et al., 2014; Botički et al., 2018; Mioduser & Levy, 2010; Wang, et al., 2014). The requirement of coding proficiency did not allow for baseline assessment of complete novices or include measurement of problem-solving skills prior to learning to code. We created an assessment instrument (*TechCheck*) that employs unplugged challenges (Relkin et al., 2020). The term "unplugged" originates from a tradition of teaching computer science through exercises that do not involve coding or computers. Unplugged challenges exercise the same type of logic and thought processes that are involved in computer programming (Bell & Vahrenhold, 2018). Unplugged activities are learning challenges such as games or puzzles that are often kinesthetic in nature, carried out in groups, and do not require prior experience with coding or computers. Unplugged activities allow children lacking access to computers to learn computer science concepts (Rodriguez, 2015; Curzon 2013; Nishida et al. 2009*). TechCheck* permits assessment of unplugged problem-solving skills before, during and after young children learn to code. The assessment allows for researchers and educators to examine how coding curricula promotes other domains of thinking.

Method

Second grade students from the Norfolk, Virginia public school system participated in the CAL-KIBO research program. The CAL-KIBO curriculum involves instruction and play with the KIBO educational robot, a developmentally appropriate, programmable screen-free robot designed to engage young children (ages 4+) in playful and creative coding. The CAL-KIBO curriculum consists of twelve lessons delivered in two hours of instruction per week over six weeks. CAL-KIBO introduces programming as a form of language and explicitly connects various programming concepts to parallel concepts in natural language and literacy (Hassenfeld, et al., 2020).

Participants in the present analysis included n= 271 second grade students from eight elementary schools enrolled in the CAL-KIBO program. This subset was selected based upon completion of *TechCheck* at three time points (baseline, midpoint, end point) as well the platform specific coding and CT skills assessment (*TACTIC-KIBO*) at the study endpoint. Only neurotypical students within this sample were included in the current analysis. The subset of students who completed the required assessments (*TechCheck* and *TACTIC-KIBO*) was relatively well-matched in terms of age, gender, and race/ethnicity to the entire cohort of students (N= 609).

Data Sources

Participating students were administered a battery of age-appropriate formative and summative assessments designed to evaluate three domains: KIBO coding proficiency (KIBO Mastery Challenges (*KMCs*), Platform Specific Coding and CT skills (*TACTIC-KIBO*) and unplugged CT and problem-solving skills (*TechCheck*). Table 1 provides more information about the instruments used in this study. Appendix 1 provides example questions of each assessment. We

received demographic information from the participating schools that was anonymized in compliance with an IRB-approved protocol.

Analytic Approach

Statistical analyses were conducted in R (Version 3.6.1, R Core Team, 2019) using R Studio version 1.2 and IBM SPSS (Version 26, 2019). Screening analyses were performed to investigate normality, linearity, and outliers. Pearson correlation coefficients were calculated to examine the bivariate relationship between all measures. Upon confirming the suitability of the data for regression analysis, we explored how the acquisition of coding skills through a coding program, CAL relates to CT and unplugged (non-coding) and problem-solving skills.

Two simultaneous regressions were carried out to examine whether the baseline *TechCheck* score (unplugged CT and problem-solving assessment) predicted the *KMC* (KIBO coding proficiency) score and the *TACTIC-KIBO* score.
A sequential regression was conducted to assess the significance and magnitude of contributions of each predictor variable (Baseline *TechCheck, TACTIC-KIBO, and KMC score)* to the end point *TechCheck* outcome.

Results

The mean age of participants was 7.57 years (*SD* = 0.56, range 7-9). 52.03% self-identified as girls. The majority of the sample identified as Black or African American (*n*=122). This was followed in frequency by students identified as White (*n*= 111), as Hispanic/ Latino/a (*n*=27), Biracial/Multiracial (*n*=19), Asian or Pacific Islander (*n*=4), and American Indian/Native American (*n*= 3). Participating schools had between 15- 80% military-connected students. The majority of students had little or no coding experience prior to the start of the study. Demographic information for the study cohort is shown in Table 2.

Participants' *TechCheck* scores improved from a mean baseline score of 11.31 (*SD* = 2.16) to a mean end point score of 12.17 (*SD*= 2.28) over the course of the six-week curriculum ($t = 4.51$, df = 538.63, $p < 0.001$). Descriptive statistics for the study measures are shown in Table 3.

The baseline *TechCheck* score significantly correlated with the coding proficiency (KMC score) $r = 0.40$, $p<0.001$ as well as with platform-specific coding and CT abilities (*TACTIC-KIBO* total score) $r=0.44$, $p<.001$. The Pearson correlations for the set of predictor and outcome variables is shown in Table 4. *TechCheck*, *TACTIC-KIBO*, and *KMC* scores were all significantly correlated, with the highest correlation coefficient obtained among the three timepoints of the *TechCheck* assessment.

We conducted two simultaneous linear regression models to assess whether baseline unplugged CT and problem-solving assessment scores (*TechCheck*) predict, respectively, the formative measure of coding (*KMC score*) and the end point measure of platform-specific coding and CT skills *(TACTIC-KIBO*) score. The simultaneous regressions indicated that baseline *TechCheck* score significantly predicted the *KMC* score ($\beta = 0.21$). Seventeen percent of the variance in *KMC* score was attributable to the baseline *TechCheck* measure $F(1, 98) = 20.65$, $p<.001$. The

baseline *TechCheck* score significantly predicted the endpoint *TACTIC-KIBO* score (β = 0.72). Nineteen percent of the variance in the *TACTIC-KIBO* score was explained by the *TechCheck* baseline score $F(1, 269) = 62.99$, $p < .001$.

We conducted a sequential regression to examine the magnitude of the contributions of each predictor variable (Baseline *TechCheck, TACTIC-KIBO,* and *KMC* score*)* to the end point *TechCheck* outcome $F(3, 96) = 15.55$, $p < .001$. About 31% of the improvement in unplugged CT and problem-solving (*TechCheck*) following CAL-KIBO were attributable to endpoint platform-specific coding and CT skills (*TACTIC-KIBO*, β = 0.199, $p<.001$) and inversely predicted by baseline "unplugged" problem solving skills (β = -0.65, $p<.001$). KIBO coding knowledge (*KMC* score) was not a significant contributor (β = 0.33, $p = .077$).
Post analysis screening was conducted and no influential cases were found. The assumption of multicollinearity and homoscedasticity was met. Inspection of histograms and p-p plots showed the standardized residuals were approximately normally distributed.

Conclusion

Participation in the CAL-KIBO was associated with improvement in unplugged problem-solving abilities. Improvement in unplugged problem-solving was related to the proficiency in coding and CT skills at the end of the curriculum. A measure of coding proficiency alone that does not take into account CT skills (*KMC*s) more weakly predicted improvement in unplugged problem-solving. This suggests that coding can be a vehicle for developing better problem-solving skills, particularly in children who generalize the knowledge gained in learning to code into computational thinking skills.

It is noteworthy that the measure of unplugged problem-solving ability in this study (*TechCheck*) showed improvement with exposure to CAL-KIBO even though the curriculum did not explicitly include the types of challenges included in *TechCheck*. This may indicate that the observed problem-solving improvements were a consequence of knowledge and skills gained in the course of learning to code and not a function of explicit instruction in solving unplugged challenges.

The performance of *TechCheck* in this large study cohort is encouraging. The instrument was easily administered and scored making it suitable for use in research and educational settings. *TechCheck* may be useful for identifying students with computational talent or potential need for extra support prior to initiating coding instruction.

Limitations

Our sample consisted of only second graders from one school district. There was a high percentage of military connected students whose family members were subject to mobilization. Consequently, only 52.22% of the students enrolled in the parent study were present for all three timepoints of the *TechCheck* assessment.

The apparent inverse relationship observed between baseline *TechCheck* scores and improvement in endpoint *TechCheck* scores was unexpected. Examination of the psychometric properties of *TechCheck* (Relkin et al., 2020) revealed evidence of a slight ceiling effect among

second graders. Consequently, students with lower baseline scores had more room for improvement than those with baseline scores closer to the assessment's ceiling. This is the likely explanation for the observed inverse relationship. Future studies in progress in younger children should shed further light on this hypothesis.

The current analysis did not include a comparison group without exposure to the CAL curriculum. Learning effects from repeated exposure to the assessments cannot be ruled out. However, no significant learning effect was observed in other analyses using *TechCheck* serially (Relkin et al., manuscript in preparation)

The CAL-KIBO curriculum differs from other coding educational initiatives in several ways including its emphasis on literacy concepts. The type of link made between coding and literacy is somewhat analogous to the link between coding and unplugged activities. The CAL-KIBO approach could therefore have influenced the *TechCheck* outcome apart from the effects of learning to code. Further studies that include comparisons of the CAL approach to other methods of teaching coding could be informative.

Significance

Coding is a valuable skill that can provide additional benefits for child development when it promotes improvements in other domains of thinking. This study is among the first to explore how teaching young children to code can improve other problem-solving skills. Translation of coding knowledge into more general CT skills may permit young children to address a broader range of problems than just those encountered in programming. We plan to further examine the elements of the coding curriculum that encourage this type of generalization of learning in the hope of improving future teaching strategies. Our current findings have implications for future studies as well as for the design of coding and other educational initiatives for young children.

References

Barlow-Jones, G., & Van der Westhuizen, D. (2017). *Problem Solving as a Predictor of Programming Performance*. 209–216. https://doi.org/10.1007/978-3-319-69670-6_14

Bell, T., & Vahrenhold, J. (2018). CS Unplugged—How Is It Used, and Does It Work? *Adventures Between Lower Bounds and Higher Altitudes* (pp. 497-521). Springer, Cham. https://doi.org/10.1007/978-3-319-98355-4_29

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. https://doi.org/10.1016/j.compedu.2013.10.020

Botički, I., Kovačević, P., Pivalica, D., & Seow, P. (2018). Identifying Patterns in Computational Thinking Problem Solving in Early Primary Education. *Proceedings of the 26th International Conference on Computers in Education*. Retrieved from https://www.bib.irb.hr/950389?rad=950389

Curzon, P. (2013). cs4fn and computational thinking unplugged. In *Proceedings of the 8th workshop in primary and secondary computing education* (pp. 47-50). ACM.

Fraillon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2018). International Computer and Information Literacy Study: ICILS 2018: Technical Report. Retrieved from https://www.springer.com/gp/book/9783030193881

Grover, S., & Pea, R. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, *42*(1), 38–43. https://doi.org/10.3102/0013189X1246305

Hassenfeld, Z. R., Govind, M., de Ruiter, L. E., & Bers, M. U. (2020). If You Can Program, You Can Write: Learning Introductory Programming Across Literacy Levels. *Journal of Information Technology Education: Research*, 19, 65-85. https://doi.org/10.28945/4509

Koulouri, T., Lauria, S., & Macredie, R. D. (2015). *Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches.* Association for Computing Machinery. https://doi.org/10.1145/2662412

Lishinski, A., Yadav, A., Enbody, R., & Good, J. (2016). The influence of problem solving abilities on students' performance on different assessment tasks in CS1. In *Proceedings of the 47th ACM technical symposium on computing science education* (pp. 329-334).

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? Computers in Human Behavior, 41, 51–61. https://doi.org/10.1016/j.chb.2014.09.012

Mioduser, D., & Levy, S. T. (2010). Making Sense by Building Sense: Kindergarten Children's Construction and Understanding of Adaptive Robot Behaviors. *International Journal of Computers for Mathematical Learning*, 15(2), 99–127. https://doi.org/10.1007/s10758-010-9163-9

Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS unplugged design pattern. ACM SIGCSE Bulletin, 41(1), 231-235.

Nowaczyk, R. H. (1984). The relationship of problem-solving ability and course performance among novice programmers. *International Journal of Man-Machine Studies*, *21*(2), 149–160. https://doi.org/10.1016/S0020-7373(84)80064-4

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from https://www.R-project.org/

Relkin, E., de Ruiter., L., Bers, M.U. (2020). TechCheck: Development and Validation of an Unplugged Assessment of Computational Thinking in Early Childhood Education.

Journal of Science Education and Technology. https://doi.org/ 10.1007/s10956-020-09831

Rodriguez, B. R. (2015). *Assessing computational thinking in Computer Science Unplugged activities* (Doctoral dissertation, Colorado School of Mines. Arthur Lakes Library).

Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. In Computational Thinking Education (pp. 79-98). Springer, Singapore. https://doi.org/10.1007/978-981-13-6528-7_6

Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, *18*, 47-58. https://doi.org/10.1016/j.ijcci.2018.06.004

Wang, D., Wang, T., & Liu, Z. (2014). A Tangible Programming Tool for Children to Cultivate Computational Thinking [Research article]. https://doi.org/10.1155/2014/428080

Wing, J. (2006). Computational Thinking. CACM vol. 49, no, 3. March 2006, pp. 33-36. https://doi.org/10.1145/1118178.1118215

Wing, J. (2011). Research notebook: Computational thinking—What and why? The Link Magazine, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from: https://www.cs.cmu.edu/link/research- notebookcomputational-thinking-what-and-why

Zhang, L., & Nouri, J. (2019) A systematic review of learning computational thinking through Scratch in K-9. Computers in Education, 141, 103607. https://doi.org/10.1016/j.compedu.2019.103607

Table 1. Description of Study Measures

Measure	Description	Specifications
TechCheck	Formative assessment using "unplugged" (non-coding) tasks to measure CT related problem-solving abilities	● 15 questions ● Multiple-choice format ● Designed for children ages 5-8 ● Average of 13.40 minutes to administer ● Score range 0-15; higher scores indicate more correct ● 2 alternate forms available ● Validated against expert assessment of CT abilities
TACTIC-KIBO (Tufts Assessment of Computational Thinking in Children - KIBO version)	Summative assessment of platform-specific coding and CT abilities in seven domains: Algorithms, Modularity, Hardware/Software, Control Structures, Debugging, Representation, and Design Process	● 28-questions ● Multiple-choice format ● Designed for children age 5-8 ● 30-40 minutes to administer (with intermission) ● Total score range 0-28; higher scores indicate more correct ● 4-level composite rating of CT proficiency ● 2 alternate forms available ● Validated against expert assessment of CT abilities
KIBO Mastery Challenges (*KMC*s)	Formative assessment of programming concepts specific to the CAL-KIBO curriculum as "checks of learning". *KMCs* assess students' understanding of the semantics and syntax of programming without requiring them to solve problems. KMCs were given at 4 time points during and after the coding program.	● Multiple-choice format ● Designed for children ages 5-8 ● 4 assessments, 6 questions per assessment for a total of 24 questions ● High Interrater reliability ● Difficulty index was calculated for each question and summed into a weighted Difficulty Composite Score ● The total points of *KMC*s ranged from 0-6.25.

Table 2. Demographics of all students who were included in the analyses

Total N		271
Self- reported age	Mean	7.57
	SD	0.56
	Range	7-9
Self- reported gender	Girl	141 (52.03%)
	Boy	127 (46.86%)
	Rather not say	3 (1.11%)
Race/ ethnicity	Black/African American	122
	Hispanic or Latino/a	27
	Biracial/ Multiracial	19
	Asian or Pacific Islander	4
	White	111
	American Indian/Native American	3
	Not Available	12

* Race/Ethnicity are not mutually exclusive

Table 3. Descriptive Statistics for Continuous Variables

n=217	*Mean (SD)*	*Min*	*Max*	*Skewness*	*Kurtosis*
TechCheck Baseline	11.31 (2.16)	4	15	- 0.53	0.25
TechCheck Midpoint	11.95 (2.06)	4	15	-0.68	0.45
TechCheck EndPoint	12.17 (2.28)	3	15	-1.14	1.54
TACTIC-KIBO	18.79(3.59)	4	26	-0.51	0.46
KMC	3.44(1.05)	1.59	6.25	0.3	-0.61

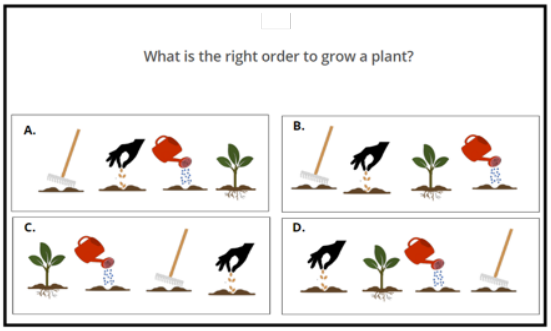
Table 4. Pearson Correlations Among Continuous Variables

Variable	*1*	*2*	*3*	*4*	*5*
1. TechCheck Pre time point	-				
2. TechCheck Mid time point	*0. 61****	-			
3. TechCheck End time point	0.58***	*0.57****	-		
4. TACTIC-KIBO total score	0.44***	0.51***	0.49***	-	
5. KMCs	*0.40****	0 .52***	0 .43***	0.48***	-

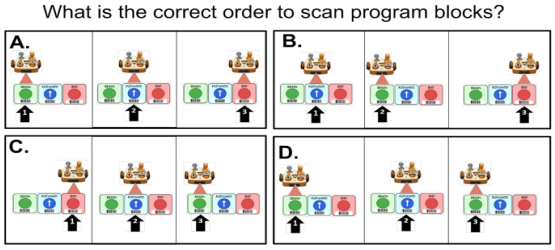
*** *Correlation is significant at the p<.001 level*

Appendix 1: Sample Assessment Items

TechCheck



TACTIC-KIBO



KMCs