Designing a Blended, Middle School Computer Science Course for Deeper Learning: A Design-Based Research Approach

Shuchi Grover, SRI International, shuchi.grover@sri.com
Roy Pea, Stanford University, roypea@stanford.edu

Abstract: As computer science (CS) thunders its way into K-12 classrooms, lack of teachers to teach CS and pedagogically sound computing curricula remain significant challenges. This paper describes a design-based research (DBR) approach to create and refine a 7-week middle school introductory CS course—*Foundations for Advancing Computational Thinking (FACT)*—informed by research in the learning sciences and computing education, and designed for blended in-class learning using online materials created on the OpenEdX platform. The paper shares FACT's success in achieving a balanced pedagogy to meet its goals for deeper learning of algorithmic concepts, and identifies areas for improvements.

Introduction

In a world infused with computing, 'computational thinking' (CT) skills are seen as key for all citizens in the digital age, not only computer scientists (Wing, 2006, Grover & Pea, 2013). President Obama's 2016 "Computer Science For All" initiative has paved the way for expansion of CS to K-12. However, few structured curricula exist for middle and elementary school levels. The middle school years are formatively central for cognitive and social development in K-12 schooling especially regarding future engagement with STEM fields (Tai et al., 2006). Middle school experiences should thus make students amenable to diverse future opportunities. Unfortunately, middle schools, like other levels of K-12, face an acute shortage of teachers to teach introductory computing curricula.

Developing well-designed online curricular materials makes possible accelerating scaling to wider audiences of students and teachers. Although recent years display growth of online CS materials on venues such as Khan Academy and Code.org, their success for development of deeper, transferable CT skills is yet to be empirically validated, and they also appear to lack robust assessment measures. Recently advancing MOOC platforms could serve this crucial need in K-12, but to be effective for younger learners, online curricula for middle school students would have to be consciously designed for active learning and engagement. What should a first-of-its-kind introductory CS course created on a MOOC platform for blended in-class learning in middle school look like? What is the best balance of pedagogies, online and offline, individual and collaborative, open-ended and directed activities for "deeper learning"? Informed by past research in the learning sciences and computing education, this research adopted an iterative process to design and refine a blended middle school introductory CS course, *Foundations for Advancing Computational Thinking* (FACT), and to empirically answer questions on the development of deeper, transferable CT skills among teens.

This paper focuses on the use of design-based research (DBR) as a methodology to put learning theory to work for designing the learning environment for the online FACT materials and the blended in-class learning experiences as well as investigating curriculum sequences, instructional approaches, activities, and assessments. It describes how initial designs of the intervention that represented "embodied conjectures" (Sandoval & Bell, 2004) of the researchers were refined over two iterations of DBR with help from stakeholders as active "design partners". As such, the quantitative data analyses of learning outcomes are not the main thrust of this paper. They are described in much more detail elsewhere (Grover, Pea, & Cooper, 2016).

Design-based research methodology for designing and refining FACT

The driving goal of our research was to design a curriculum to prepare middle school learners for CT. The first step in establishing the broad viability and usefulness of a curricular intervention to achieve desired outcomes as described above was to design and test how well it worked in a *real classroom setting*. Specifically, how effective is the FACT curriculum for helping middle school students develop awareness, positive attitudes and interests towards the discipline of computer science, and a deeper understanding of foundational computing concepts that can transfer to future computing experiences? Designing and studying FACT in context was arguably a 'Type 1' translational research effort (Pea, 2010), and thus benefited from a DBR approach, where the key stakeholders typically involved are teachers, learners, learning scientists, subject matter experts, and technology developers. The types of questions typically answered through such research are: (a) *What do students learn from this design?* (b) *How do students learn from this design?* (c) *What do problems in learning or implementation suggest about re-design of the intervention?*

DBR is increasingly being embraced for research in the learning sciences, instructional design, and curriculum development. It includes "testing theoretical accounts of learning, cognition, and development by using the theory to design or engineer instruction and conducting empirical studies of the instruction in a naturalistic setting" and it responds "to the need to study complex interventions that include a range of intentionally-designed features and materials such as curriculum sequences, technological tools, social norms, instructional approaches, activity structures, and cognitive assessments in complex settings" (Bell, Hoadley, & Linn, 2004). DBR addresses the systemic, complex nature of education by researching curricular interventions in real-life settings, often involves stakeholders such as teachers and learners in the design process, and aligns particularly well with the goal of promoting inquiry in STEM courses as well as technology-enhanced learning environments (Barab & Squire, 2004).

A key feature of DBR studies is that they are typically iterative in nature, and involve refinement of the intervention. The initial tentative set of design considerations embodies the conjectures of the researcher(s)-designer(s) that are then revised depending on their successes and challenges experienced in practice (Edelson, 2002). Bell et al. (2004) suggest that DBR tends to include compelling comparisons in which two forms of the innovation are enacted under otherwise similar conditions and variations in the innovation used during such compelling comparisons test hypotheses about learning embodied in the designs. DBR methodologies specifically targeting design of learning environments also call for designs that are *pragmatic* and *grounded* with respect to theories of how people learn and the specific contexts within which the technologies are implemented (Wang & Hannafin, 2005). The goal of design research is also to help develop domain theories, design frameworks and/or design methodologies.

Stakeholders as 'design partners' and iterative research design

The novelty of the curricular materials and the online platform in this context necessitated drawing from learning theory and past research on children and programming as a foundation and starting point for the **initial** designs of FACT. Using ideas as advocated in DBR for getting guidance from stakeholders on what makes sense for middle school children using a curriculum such as this one, we adopted an iterative process to refining the curriculum. Given the newness of MOOC platforms and that this effort was a first-of-its-kind blended learning course using such a platform for a K-12 setting, the idea of involving the classroom teacher and learners as 'design partners' seemed particularly appropriate. Adopting a DBR methodology thus involved testing the curriculum *in context*—in a real middle school classroom setting—and gathering detailed feedback from students and the classroom teacher, the key stakeholders who were 'design partners' in this endeavor. The original curriculum before the first iteration represented the embodied theoretical conjectures about how middle school learners can best achieve the desired outcomes, and as such, the curriculum carried "expectations about how designs should function in a setting." (Sandoval & Bell, 2004).

Guided by findings from preliminary explorations and the design-based research (DBR) approach, empirical investigations were conducted over two iterations (hereafter referred to as Study1 and Study2) of teaching FACT in a public school classroom. In keeping with DBR philosophy, in both iterations, frequent feedback was sought from students during the intervention in addition to extensive feedback during and after the course via surveys. Students were reminded often by the classroom teacher that in addition to being learners, they also had a role in this research as 'design partners' and *that their inputs and feedback were crucial to refine the course for future students who would learn from an improved course that will have incorporated the students' ideas for improvement.* The lead researcher met with the teacher several times in between the first and second iteration to go over student feedback from Study1, and to solicit her ideas for how those suggestions could translate into course features for the online materials and blended course.

The two iterative studies involving the use of FACT in a middle school classroom investigated the research questions: RQ1- What is the variation across learners in achieving desired outcomes through FACT, specifically the learning of algorithmic flow of control– (a) serial execution (b) looping constructs and (c) conditional logic, and what factors influence this variation? RQ2- How well does the curriculum promote an understanding of algorithmic concepts that goes deeper than tool-related syntax details as measured by PFL assessments? RQ3- How do students' perception of the discipline of CS change as a result of FACT?

Teaching the curriculum face-to-face first (in Study1) *without the constraints of the online medium afforded a focus on the pedagogical content knowledge or PCK* (Shulman, 1987) *designs as well as design of assessments and surveys for gathering feedback used in the curriculum*. Furthermore, using the first iteration to pilot a portion of the curriculum as an online course on a MOOC platform in Study1 helped researchers to observe the classroom learning experience and elicit student as well as teacher feedback that informed subsequent refinements of the initial set of design and pedagogical assumptions about the use of the online version of FACT in a blended classroom setting. Specifically, the researchers set out to examine how the initial curriculum and learning experience that embodied conjectures of the researchers/designers would be enacted in a classroom setting. In addition to questions on how students learned specific computing concepts as measured by pre-post tests, there were design questions for consideration. Our (representative) list of questions about the design and curriculum that we aimed to answer through DBR was: (1) how should the pedagogical balance between open-ended exploration, guided inquiry, instruction and directed activities be achieved? (2) how should online and offline activities and topics be sequenced? (3) what should be the ideal length of instructional videos? (4) which Scratch programming projects are the most fun for learners and which are the most difficult? (5) how can authentic activities be incorporated into the curriculum that attend to learner agency while still keeping directed programming assignments that address specific learning goals? (6) which videos are most effective, and which the least, in raising students' interest in CS? why? (7) what steps could be taken to make the online modules more engaging? (8) how should students with varying abilities be accommodated in a blended course that requires students to move every week to a new unit?

DBR with a difference

It should be noted that in designing a curriculum with clear *a priori* operationalized outcomes (represented by designed measures) and refining it to create an optimal learning experience that also results in learners achieving those desired outcomes, this approach could be viewed as somewhat problematic in some researchers' views of DBR. Engeström (2009), in particular, takes issue with the linear fashion of research that many forms of DBR efforts have adopted such as Collins, Joseph & Bielaczyc (2004) that start with researchers determining the principles and goals and going through subsequent phases of refinement leading to completion or perfection. A methodological goal of this research too was to refine the FACT in order to best achieve desired outcomes, often relevant to building an "outcomes" domain theory (Edelson, 2002). There is, however, resonance in the assertions of von Hippel and Tyre (1995) that Engeström (2008) cites to make his point about what he considers the true nature of DBR– "one can never get it right, and that innovation may best be seen as a continuous process, with particular product embodiments simply being arbitrary points along the way."

Methods

Participants and procedures

Empirical studies were conducted in a Northern California public middle school classroom. Two iterations (Study1 and Study2) of the design-based research (DBR) were conducted with two different cohorts in the same 'Computers' elective class that met for 7 weeks, with four 55-minute periods per week. The samples comprised 7th and 8th grade students (Table 1). In both iterations, approximately a fifth of the class comprised students who had been placed in the elective class by the school counselors. These students happened to be English language learners or students with other learning difficulties. Unfortunately, since this was an elective class, these students did not get the same specialist supports they received in core subject classes. The classroom teacher, who did not have a background in CS or programming, was present in the classroom at all times assisting with classroom management and "learning right alongside the students."

Table 1: Student Samples in Study1 & Study2

		Count by Gender		Count by Grade		Count in Sp. Programs	
Study	Mean Age	Male	Female	Grade 7	Grade 8	ELL	Special Ed.
1	12.9	21	5	15	11	4	2
2	12.3	20	8	16	12	3	1

In Study1, the course was taught face-to-face by the lead researcher. ***One unit in Study 1 was piloted on the MOOC platform with face-to-face instruction replaced by videos and interspersed with automated quizzes.*** Extensive feedback was sought from learners on their experiences with the online unit as a precursor to creating online curricular materials for the entire curriculum for Study2. Study2, conducted in the same classroom with a new cohort, used a designed blended learning experience with online FACT materials on Stanford's OpenEdX platform (roughly 50 videos, 1-5 minutes in length), quizzes, and Scratch activities to be done individually or collaboratively that were interspersed through the course. In addition, several refinements were made to the curriculum based on experiences and student feedback in Study1. The studies were conducted during two semesters in 2013 that included visits to the classroom preceding and following the 7-week long FACT intervention for IRB permissions, pre-post tests, as well as wrap up of final projects and presentations.

Data measures

Table 2: Instruments used to capture key data measures

Instrument	Pre	Post	Source(s)
Computational Knowledge Test	X	X	Ericson and McKlin (2012); Meerbaum-Salant et al. (2010); Zur Bargury et al. (2013)
Preparation for Future Learning (PFL) Test		X	Designed (Inspired by Schwartz and Martin (2004))
Prior Experience Survey (programming experience and technology fluency)	X		Adapted from Barron (2004)
CS Perceptions Survey	X	X	Ericson and McKlin (2012)
Online Learning Experience Survey (Study 2)	X	X	Designed (Inspired by Barron (2004))
FACT Experience survey		X	Designed (for getting student feedback)
Classroom Observations			During the interventions for DBR

FACT curriculum and pedagogy

The 7-week FACT curriculum (Table 3) included topics that were considered foundational and appropriate for middle school students and focused on algorithmic thinking required for learning programming. The curriculum design effort was guided by goals for deeper learning (Pellegrino & Hilton, 2013), which attend to the development of cognitive abilities through mastery of disciplinary learning, in addition to interpersonal and intrapersonal abilities. The pedagogy for CT followed a scaffolding (Pea, 2004) and cognitive apprenticeship (Brown, Collins, & Newman, 1989) approach. It involved the use of (worked) examples to model solutions to computational problems in a manner that revealed the underlying structure of the problem, and the process of composing the solution in pseudo-code or in the Scratch programming environment. Academic language and computing vocabulary were used throughout this scaffolding process. The course emphasized "learning by doing" for students through a mix of directed assignments as well as meaningful, open-ended projects (Barron & Darling-Hammond, 2008) in Scratch including a substantive culminating project. Frequent low-stakes and high frequency multiple-choice assessments were designed to keep learners deeply engaged with the course's content and understanding goals, and to provide feedback for both learners and the teacher. 'Preparation for future learning' (PFL; Schwartz, Bransford, & Sears, 2005) for text-based computing contexts was mediated through expansive framing pedagogical moves (Engle et al., 2012) and providing learners opportunities to work with analogous representations (Gentner et al., 2003) of the computational solutions—plain English, pseudo-code and Scratch programs (Grover, Pea, & Cooper, 2014b). Summative assessments included a specially designed PFL test in addition to other performances of CT understanding. FACT also consciously engaged with students' narrow perceptions of CS to help them see computing in a new light through a curated video playlist that illuminated computing as a creative, problem-solving discipline with applications in many real-world contexts (for details see: Grover, Pea, & Cooper, 2014a). The goal of DBR was to use the data measures to test the pedagogy and designed curriculum & assessments to achieve desired outcomes.

Table 3: Curriculum Sequence for FACT

Unit 1	Computing is Everywhere! / What is CS?
Unit 2	What are algorithms & programs? Computational solution = Precise sequence of instructions
Unit 3	Iterative/repetitive flow of control in a program: Loops and Iteration
Unit 4	Representation of Information (Data and variables)
Unit 5	Boolean Logic & Advanced Loops
Unit 6	Selective flow of control in a program: Conditional thinking
	Final Project (student's own choice; could be done individually or in pairs)

Refinements to learning environment after Study 1

Key improvements to Study1 were incorporated in the FACT online/blended version for Study2:

1. Videos were shortened from ~8-10 mins to ~1-5 mins in Study 2 (to heighten learners' engagement).
2. More time was devoted to loops and variables (since they struggled with those topics most).
3. Small modifications in content sequence that made more sense.
4. Addition of Scratch window below video for students for greater interactivity.
5. More games and creative artifacts in Scratch Assignments (based on student feedback).
6. More engaging corpus of '*Computing is Everywhere*!' videos. (Additional videos to show after week 1)

7. More fun worked examples in Scratch ('Pong' for conditionals; '4-quadrant' art for Boolean logic).
8. Final project of choice (in pairs or individual), whole-class demo, and online showcase of projects.
9. A 'Word Wall' tab for computational terms.
10. Additional thought questions as well as online space for contextual questions (below each video).
11. Improved survey question design for soliciting student feedback.

Analysis and results

In order to answer the three research questions, data were analyzed separately for each study using mixed method techniques. The pre-to-posttest effect size (Cohen's *d*) on the CT test was ~2.4 in both studies. Although students scored an average of 65% on the PFL test, there was evidence of understanding of algorithmic flow of control in code written in a text-based programming language. Most of the PFL questions involved loops and variables—topics that students had the most difficulty with in the computational learning posttest. Regression analyses suggested that the curriculum *helped all students regardless of prior experience as measured by the self-report survey, however the pretest score was found to be a significant predictor for both the posttest and the PFL test*. Responses to the perceptions of computing question were analyzed using a mix of qualitative coding and quantitative methods. They revealed a significant shift from naïve "computer-centric" notions of computer scientists to a more sophisticated understanding of CS as a creative, problem-solving discipline.

Comparative analysis of Study 1 vs. Study 2

Computational learning and PFL (Transfer)

On the main outcomes of interest, namely the CT posttest score and the PFL test score, there was no significant difference among the students in Study 1 and Study 2. The pretest scores in the two studies were also not statistically different. *The learning gain, however, calculated as the difference between the posttest and pretest scores was significantly higher in Study 2 on both the t-test and the non-parametric Mann Whitney test.* The PFL test performances were also comparable and the difference between the average PFL scores in Study1 and Study2 was not significant. These results and score breakdowns by CT construct are shown in Tables 4 and 5. It should be noted that most of the questions on loops in the posttest also included conditionals and serial execution, in addition to variable manipulation—a topic with which students struggled in both studies. For the questions involving loops, students thus needed a good understanding of how different computational constructs come together in a program, and also of variable manipulation within loops. Both these aspects are particularly difficult for novice programmers (Pea, 1986; Soloway, 1986; Spohrer & Soloway, 1986).

Table 4: Student Samples in Study1 & Study2

	Study 1		Study 2					
	N	Mean (SD)	N	Mean (SD)	t	p < |t|	z	p < |z|
Pretest	24	36.33 (18.19)	28	28.06 (21.18)	1.5	0.14	1.76	0.08
Posttest	26	78.58 (17.08)	28	81.60 (21.24)	-0.58	0.56	-1.26	0.21
Learning Gain	24	43.08 (12.17)	28	53.07 (18.34)	-2.34	0.02	-2.46	0.01
PFL Test	25	63.37 (28.86)	27	65.07 (26.47)	-0.22	0.82	-0.08	0.93

Table 5: Student Samples in Study1 & Study2

Variable	Study 1 Mean (SD)	Study 2 Mean (SD)	t-Stat	p	Z-Score	p
Overall	78.6 (17.1)	81.6 (21.2)	-0.6	0.56	-1.3	0.21
By CS Topic						
Serial Execution	97.4 (13.1)	91.1 (20.7)	1.4	0.18	1.6	0.12
Conditionals	84.5 (19.0)	84.9 (20.5)	-0.1	0.94	-0.4	0.72
Loops	74.1 (21.9)	77.2 (26.3)	-0.5	0.64	-1.1	0.29

Perceptions of computing

There was no statistical difference in students' pre-post attitudes and interests in computing in either study. Such a ceiling effect is not uncommon when learners self-select into the intervention, and enter with high levels of interest and motivation. However, students in Study2 performed better than those in Study1 on growth of students' perceptions of computing, tested mainly through pre-post responses to the question "*In your view, what do computer scientists do?*" There were no significant differences between Study1 and Study2 in any of the key coding categories that were used to code the responses. Since DBR-inspired refinements in the Study2 materials were aimed at tackling learners' awareness and understanding of computer science, some improvements in student performance were to be expected. For this reason, we also tested the *a priori* hypothesis that in assessing the more fine-grained aspects of the responses (***richness*** and ***length*** of answers to "*In your view, what do computer scientists do?"*) students in Study2 would do better than those in Study1. Statistical analyses confirmed this hypothesis: the differences across the two studies (as measured by t-tests and Rank Sum tests) were significant, with students in Study2 performing significantly better. The difference in the post-course responses as measured by the number of meaningful codes (richness) was significant at $p<0.01$, and the increase in response length from Study1 to Study2 was also significant at $p<0.001$ (Table 6).

Table 6. *"What do computer scientists do?", Comparison of Responses in Study 1 & Study 2*

Variables (N = 54)	Study 1 Mean	Study 2 Mean	t	P(|T| > |t|)	z	P(|Z| > |z|)
Length, Before	80.8 (59.5)	57.0 (38.1)	1.7	0.089	1.4	0.161
Richness, Before	0.4 (0.9)	1.0 (1.2)	-2.2	0.03	-2.4	0.016
Length, After	64.0 (39.1)	181.6 (106.9)	-5.6	< .001	-4.8	< .001
Richness, After	2.0 (0.8)	3.0 (1.2)	-3.4	0.001	-3.0	0.003

Final project, presentation and interview as performance assessments in Study 2

Study1 did not have a formal project at the end of the intervention, as the weeks following FACT were disrupted by end-of-year activities. However, the success of the final open-ended project for the few children that did it informally prompted a decision to add a 7th week to FACT in Study2. After the six weeks of the FACT curriculum ended, students formally worked on a culminating project that involved designing and programming a game of their own choosing in Scratch that they then presented at a whole class 'Project Expo Day'. Sometimes they also projected their code to show how certain aspects of the game were programmed. At other times, they called on their peers to demonstrate games that required two or three players. All the final projects were also uploaded to an online project "showcase" (a Scratch 'studio'), a resource enabling students to see all the projects, and play games created by their classmates. The following week, students continued to fix bugs in their projects, created short 'user manual'-styled instructions to describe game play, and documented the final project experience in a form adapted from the *Starting from Scratch* curriculum (Scott, 2013) that was provided to each student. Most importantly, students spent time playing with each other's games (or their own). In their post-survey responses, students were asked to vote the **three** Scratch projects that were "the most challenging", "the most fun" and "the least fun" among all the Scratch projects that they had completed. The final project was ranked highest on "the most fun" and also the "the most challenging".

In contrast to the de-contextualized posttest with questions that involved comprehending Scratch code and answering related questions, the final project was a more meaningful form of performance assessment (Barron & Darling-Hammond, 2008). It embodied learner agency and students felt a sense of accomplishment and pride as they presented their projects to the class and received cheers of praise from their peers. Most importantly, it seemed to work well even for the students who performed poorly on the posttest, although their projects used fewer complex structures than those of students who performed better on the posttest.

Discussion

As these results reveal, the iterative refinements driven by DBR resulted in the blended version of FACT using OpenEdX worked as well (if not slightly better on some metrics) as the face-to-face version of the FACT curriculum. Based on success metrics for an online *replacement* course (Means, et al., 2010), online/blended FACT was successful since learners did at least as well as those in the face-to-face intervention. This result is even more encouraging when we take into account students' preferences for face-to-face learning over online learning (as revealed in pre and post surveys in Study2).

It should be noted that the improvements in Study2 were observed on the aspects of the FACT learning designs that had been refined after the first iteration of this DBR investigation. They also targeted ways in which students could better appreciate the ideas behind the '*Computing is Everywhere!*' unit of the curriculum. The DBR effort thus resulted in an improved curriculum in those aspects where re-design was enacted.

Reflections on future improvements

Despite allotting more time in Study2 for hard-to-learn concepts such as variables and loops, learners still struggled with those ideas, especially those with poor math preparation. This finding suggests that perhaps we need to rethink our pedagogy for introducing those ideas to reach diverse learners. The results also suggested that for a balanced set of questions with robust construct validity, the PFL test should assess learners on the individual concepts taught—serial execution, variables, conditionals and loops—rather than only loops and variables as was the case. Additionally, many items on the pre-post surveys, pre-posttests, quizzes, and the PFL test required students to read non-trivial amounts of text—a real challenge for ELL students. Such students often require the help of aides in core subject classrooms–help that this classroom did not have as this was an elective course. Clearly the curricular materials would have to become less dependent on English, perhaps through images or simpler scenarios. Lastly, although the online FACT course designed for blended learning could be considered a success per the criteria guiding this research, classroom observations and student feedback suggest that the FACT could have incorporated more (guided) exploration and learner agency. Lastly, some students clearly would have benefited from more time; having to stick to the schedule of the research study meant that some students lagged behind in the course. This needs to be improved. Congruent with methodologies of DBR, the current version of FACT is still a work-in-progress. It is hoped that teachers who use this course can address some of these limitations in ways that work for them and their students.

DBR lessons and future work

What takeaways does this DBR effort provide to the field? Edelson (2002) contends that while the lessons of individual design effort are often restricted to the particular design and the individuals involved in it, DBR has the additional goal of developing generalizable domain theories, design frameworks and/or methodologies. Piloting a new curriculum face-to-face (with only one unit online) first to focus on examining PCK, activities and assessments without the online/blended modality, was a useful way of modularizing the problem space. It serves as a design methodology that has applicability in course design using MOOC platforms for blended or all-online settings. FACT design also provides strategies for incorporating active learning in MOOC courseware. The iterative design of FACT as an online course on a MOOC platform for blended in-class learning provides an example of incorporating lessons from the learning sciences in designing an effective learning environment that relies on video-based instruction, while balancing online and offline work by learners. Using inquiry to activate a richer web of mental connections, contextual discussions preceding and following videos, and mechanisms for learners to program and respond to thought questions as they watched videos made for a more active learning experience. Most importantly, FACT provides evidence for a balanced pedagogy for K-12 CS classrooms—one that incorporates guided instruction with active learning, and directed with open-ended programming projects for deeper learning of conceptual ideas underpinning algorithmic thinking and programming. Incorporating a wide range of mechanisms to learn, hone and demonstrate CT through various kinds of assessments including PFL transfer assessments (Grover et al., 2015) makes FACT a unique and innovative introductory CS curriculum. This research also exemplifies the value of DBR as a methodology to iterate on the design of such a first-of-its-kind course. In their role as "design partners", students provided extensive feedback on many aspects of the course. This was immensely valuable in tailoring worked examples and assignments so that they are more in tune with students' contexts and interests.

Future iterations involve improving on the curriculum design based on results and student feedback in Study2 and using FACT with broader audiences of middle school students and teachers across the US. Alternate pedagogical strategies will be tested for teaching topics (such as loops and variables) that learners found to be challenging, and in ways that reach all children that have varying levels of prior math preparation.

Any curricular innovation is a continuous process, and any particular version of it is simply a point along the way. This is true of FACT as well. Our learning and findings from these studies present promising directions for future research involving further improvements in FACT. FACT currently embodies a well-conceived, pedagogically robust design of a curriculum for computing in middle school that has been empirically tested for CT learning outcomes in school settings. Nonetheless, it still has room for improvements. This research represents the first two iterations of what should be seen as an ongoing systematic design-based research effort in diverse settings and with broader audiences of middle school students and teachers.

References

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *Journal of the Learning Sciences*, *13*(1), 1-14.

Barron, B. (2004). Learning ecologies for technological fluency: Gender and experience differences. *Journal of Educational Computing Research*, 31(1), 1-36.

Barron B., & Darling-Hammond, L. (2008). How can we teach for meaningful learning.? In Darling-Hammond, et al. *Powerful learning: What we know about teaching for understanding*. San Francisco: Jossey-Bass.

Bell, P., Hoadley, C. M., & Linn, M. C. (2004). Design-based research in education. *Internet environments for science education*, 73-85.

Brown, J. S., Collins, A., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, *487*.
Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design research: Theoretical and methodological issues. *Journal of the Learning Sciences*, *13*(1), 15-42.
Edelson, D. C. (2002). Design research: What we learn when we engage in design. *Journal of the Learning Sciences, 11*(1), 105-121.
Engeström, Y. (2009). The future of activity theory: A rough draft. *Learning and expanding with activity theory*.
Engle, R. A. et al. (2012). How does expansive framing promote transfer? Several proposed explanations and a research agenda for investigating them. *Educational Psychologist*, *47*(3), 215-231.
Ericson, B., & McKlin, T. (2012). Effective and sustainable computing summer camps. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 289-294). ACM.
Gentner, D., Loewenstein, J., & Thompson, L. (2003). Learning and transfer: A general role for analogical encoding. *Journal of Educational Psychology*, *95*(2), 393–408.
Grover, S., & Pea, R. (2013). Computational thinking in K–12: a review of the state of the field. *Educational Researcher*, 42(1), 38-43.
Grover, S., Pea, R. & Cooper, S. (2014a). Remedying misperceptions of computer science among middle school students. *In Proceedings of the 45th ACM SIGCSE (2014). New York, NY: ACM.*
Grover, S., Pea, R. and Cooper, S. (2014b). Expansive framing and preparation for future learning in middle-school computer science. In *Proceedings of the 11th ICLS (2014)*, Boulder, CO.
Grover, S., Pea, R., Cooper, S. (2015). Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education, 25*(2), 199-237.
Grover, S., Pea, R. & Cooper, S. (2016). Factors influencing computer science learning in middle school. *In Proceedings of the 47th ACM SIGCSE (2016)*. Memphis, TN. ACM.
Means, B., Toyama, Y., Murphy, R., Bakia, M., & Jones, K. (2010). *Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies*. US Dept of Education.
Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M., (2010). Learning computer science concepts with Scratch. In *Proceedings of ICER '10*. New York: ACM, 69-76.
Pea, R. D. (2010). *Augmenting educational designs with social learning*. In *NSF SLC PI Meeting*.
Pellegrino, J. W., & Hilton, M. L. (Eds.). (2013). *Education for life and work: Developing transferable knowledge and skills in the 21st century*. National Academies Press.
Sandoval, W. A., & Bell, P. (2004). Design-based research methods for studying learning in context: Introduction. *Educational Psychologist*, *39*(4), 199-201.
Schwartz, D.& Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction*, *22(2)*, 129-184.
Schwartz, D. L., Bransford, J. D., & Sears, D. (2005). Efficiency and innovation in transfer. In J. Mestre. (Ed.), *Transfer of learning from a modern multidisciplinary perspective* (pp. 1-51). Greenwich, CT, Information Age Publishing.
Scott, J. (2013). The royal society of Edinburgh/British computer society computer science exemplification project. *Proceedings of ITiCSE'13*, 313-315.
Shulman, L. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Ed. review*, *57*(1), 1-23.
Soloway, E. (1986). Learning to program=learning to construct mechanisms and explanations. *Communications of the ACM*, *29*(9), 850-858.
Spohrer, J. C. & Soloway, E. (1986) Novice mistakes: are the folk wisdoms correct? *Communications of the ACM*, 29(7), 624–632.
Tai,R., QiLiu, C., Maltese, A.V., & Fan,X. (2006). Planning early for careers in science. *Science*. 312(5777) 1143-1144
Wang, F., & Hannafin, M. J. (2005). Design-based research and technology-enhanced learning environments. *Educational technology research and development, 53*(4), 5-23.
Wing, J. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–36.
Von Hippel, E., & Tyre, M. J. (1995). How learning by doing is done: problem identification in novel process equipment. *Research Policy*, *24*(1), 1-12.
Zur Bargury, I., Pârv, B. & Lanzberg, D. (2013). A nationwide exam as a tool for improving a new curriculum. Proceedings of ITiCSE'13, 267-272. Canterbury, England, UK.

Acknowledgments

We gratefully acknowledge funding from the National Science Foundation [NSF-0835854, NSF-1343227], and the intellectual contributions of Profs. Stephen Cooper, Daniel Schwartz and Brigid Barron in this work.

