Design-Based Research for Deeper Learning in an Online Introductory CS Course for Middle School Students

Shuchi Grover
SRI International
&
Roy Pea
Stanford University

Please address all correspondence to:
Shuchi Grover
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025
shuchi.grover@sri.com

Symposium Paper presented at the
Annual Meeting of the American Educational Research Association
Chicago, April 16–20, 2015

Design-Based Research for Deeper Learning in an Online Introductory CS Course for Middle School Students

Shuchi Grover & Roy Pea

Abstract: As computer science (CS) makes its way into K-12 classrooms, lack of teachers to teach CS and pedagogically-sound computing curricula remain significant challenges. Well-designed online courses deployed on MOOC platforms could serve a crucial need. This paper describes our efforts to use the design-based research methodology to design a 6-week middle school course on the OpenEdX platform for blended in-class learning. Using past research in the learning sciences and computing education as a foundation as well as inspiration from 'Exploring CS' high school curriculum, this research adopted an iterative process to design and refine the course 'Foundations for Advancing Computational Thinking' (FACT, and empirically study the effectiveness of FACT for development of cognitive and non-cognitive aspects of deeper, transferable computational skills among young teens.

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, 2011). It is argued that those in possession of computational competencies will be better positioned to understand and take advantage of a world with ubiquitous computing (Grover & Pea, 2013). In middle and high schools across the US, misperceptions and lack of awareness about computer science (CS) lead to a lack of interest in this growing field. Abysmally low numbers of high school AP CS Exam takers and data from the Bureau of Labor Statistics about the growth of computing jobs highlights the chasm between opportunity and capacity in computing-related careers. Appropriate experiences with computing during the K-12 years could help address this problem and broaden the CS pipeline.

While needs of high school students are being prioritized through pilot courses such as Exploring Computer Science (ECS) and AP CS Principles, there is growing belief that experiences with computing must start at an earlier age as evidenced by the push orchestrated by organizations such as Code.org. The middle school years are formative and key for cognitive and social development in the K-12 schooling journey especially with regard to future engagement with STEM fields (Tai et al., 2006). Middle school experiences should thus make students amenable to diverse future opportunities as part of their "possible selves" (Markus & Nurius, 1986). Unfortunately, middle schools across the US face an acute shortage of teachers to teach introductory computing curricula.

Making available a robust, structured curriculum as an online course would accelerate scaling to wider audiences of students and teachers. While recent months have seen growth of CS curricula at online venues like 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. The recent explosion of MOOC platforms and courses could serve this crucial need in K-12. However, in order to be effective, and especially for younger learners, an online course for middle school students would have to be consciously designed for better engagement and active learning.

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 blend for balancing online and offline, individual and collaborative, open-ended and directed activities for active, deeper

learning? Using past research in the learning sciences and computing education as a foundation, this research adopted an iterative process to design, refine, and study an online curricular intervention—'Foundations for Advancing Computational Thinking' (FACT)—to empirically answer questions on the development of deeper, transferable CT skills among young teens.

This paper focuses on the use of design-based research (DBR) as a methodology to put learning theory to work in engineering the learning environment for the online FACT course and the blended in-class learning experiences as well as investigating curriculum sequences, instructional approaches, activities, and assessments. It describes how initial designs of our 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" (Amiel & Reeves, 2008). As such, it does not go into the details of the quantitative data analysis and only briefly describes the results.

Design-Based Research Methodology for Designing and Refining FACT

The driving goal of this research was to prepare middle school learners for computational thinking, a skill set considered crucial for all in a modern world. In an expansively framed (Engle et al., 2012) introductory CS curriculum aimed at "deeper learning" (Pellegrino & Hilton, 2013) through guided inquiry and scaffolding (Brown, Collins, & Newman, 1989; Pea, 1997) the development of foundational computational thinking skills and 'preparation for future learning' (Schwartz, Bransford, & Sears, 2005) were situated within the larger context of computer science as an engaging and creative problem-solving discipline that supports computation in diverse fields. Learner success in future engagement with computing in high school and beyond relies on how well this introductory course prepares them on the cognitive and affective dimensions of computational learning.

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 the curriculum worked to achieve desired learning goals in a real classroom setting. Specifically, how effective was FACT as a curriculum in helping students develop awareness, positive attitudes and interests towards the discipline of computer science, as well as 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'[1] translational research effort (Pea, 2010), and thus benefited from a *design-based research* approach, where the key stakeholders typically involved are learning scientists, teachers, learners, subject matter experts, and technology developers. The types of questions answered were:

- What do students learn from this design?
- How do students learn from this design?
- What do problems in learning or implementation suggest about re-design of the intervention?

Design-based research (DBR) is increasingly being embraced for research in the learning

[1] To orient the field, the Institute of Medicine's Clinical Research Roundtable developed the distinction between two types of translational research: *Type-1 and Type-2* (Woolf, 2008). Type-1 was described as "the transfer of new understandings of disease mechanisms gained in the laboratory into the development of new methods for diagnosis, therapy, and prevention and their first testing in humans." Type-2 was scoped as "the translation of results from clinical studies into everyday clinical practice and health decision making." Pea (2010) describes how the learning sciences has its own versions of Type I and Type II research.

sciences, instructional design, curriculum development and teacher professional development (McKenney & Reeves, 2014). 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" (Bell, Hoadley, & Linn, 2004). DBR responds to the systemic, complex nature of education by researching curricular interventions in the "buzzing, blooming confusion of real-life settings where most learning actually occurs, and aligns particularly well with the goal of promoting inquiry in STEM courses as well as technology-enhanced learning environments " (Barab & Squire, 2004). Bell et al. (2004) also assert, "design studies respond 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" (p. 74).

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. According to Barab & Squire (2004), DBR often involves different participants so as to bring their expertise in the analysis of the design instead of treating them solely as 'subjects' in the research. 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 and technologies 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). Given the context of this learning environment–a blended course on introductory CS and computational thinking for teens–this call is interpreted as requiring empirical inquiries that study initial "embodied conjectures" (Sandoval & Bell, 2004) in the design of FACT and assessing for their effectiveness in addition to getting (and incorporating) feedback from stakeholders. 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 taking a 'first principles' approach to designing FACT as an online course. Using ideas as advocated in DBR for getting guidance from stakeholders on what makes sense for middle school aged children using a curriculum such as this one, it seemed prudent to adopt an iterative process to designing and refining the curriculum. Given the newness of MOOC platforms and that this effort was a first-of-its-kind course to use such an online platform in a K-12 setting, the idea of involving the classroom teacher and learners, as 'design partners' seemed particularly appropriate.

Stakeholders as 'Design Partners'.

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 conjectures of the researcher-designer as the instructional designs materially embodied theoretical conjectures about how middle school learners can best achieve the desired outcomes. As "embodied conjectures", the curriculum carried "expectations about how designs should function in a

setting, and tracing how such expectations are met or unmet can refine the underlying theoretical conjecture" (Sandoval & Bell, 2004).

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) and Middleton, Gorard, Taylor, & Bannan-Ritland (2008), which 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 online FACT course in order to best achieve desired outcomes. 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."

Iterative Research Design

Guided by findings from the preliminary explorations and the design-based research (DBR) approach, empirical investigations were conducted over two iterations (hereafter referred to as Study 1 and Study 2) of teaching FACT in the year 2013 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 after the course via post-course 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 Study 1, and to solicit her ideas for how those suggestions could translate into course features for the online/blended course.

The two iterative studies involving the use of FACT in a middle school classroom investigated the following research questions:

1. What variability is there across learners in achieving desired outcomes through the FACT curriculum, specifically the learning of algorithmic flow of control– (a) serial execution (b) looping constructs and (c) conditional logic?
2. Does the curriculum promote an understanding of algorithmic concepts that goes deeper than tool-related syntax details as measured by PFL assessments?
3. What is the change, if any, in students' perception of the discipline of CS as a result of experiencing the FACT curriculum?

Teaching the curriculum face-to-face first (in Study 1) without the constraints of the online medium afforded a focus on the pedagogical content knowledge or PCK (Shulman, 1987, 1994) 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 Study 1 helped 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 the how the initial curriculum and learning experience that embodied conjectures of the researchers/designers would be enacted in a classroom setting. In addition to the questions on how students learned specific computing ideas concepts as measured by pre-post tests, there were design questions we were concerned with as well. The following list presents a representative list of design and curriculum questions–

1. Which videos from the corpus of 'Computing is Everywhere' videos do the learners enjoy the most and the least? And why?
2. How do students rate the Scratch assignment projects? Which were the 3 most fun? The 3 most difficult?
3. What sort of programming activities did the learners enjoy and learn from well?
4. How do students describe the experience of watching videos for worked examples?
5. Did the students find the videos to be too long? Too short?
6. What steps could be taken to make the online modules more engaging?
7. Were the online quizzes helpful for learning?
8. How should online & offline activities and topics be sequenced?
9. How well do paired activities work? Do students work well on their own?
10. How should students with varying abilities be accommodated in a blended course that requires students to move lock-step every week to a new unit?
11. How can authentic activities be incorporated into the curriculum that attend to learner agency?

Methods

Participants and Procedures

Empirical studies were conducted in a public middle school classroom in Northern California. 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 four days a week for 55 minute periods. The student samples comprised 7th and 8th grade students (Table 1).

In Study1, the course was taught face-to-face by the lead researcher. One unit was piloted on the MOOC platform with the face-to-face lessons replaced by video ones and interspersed with automated quizzes. Extensive feedback was sought from learners on their experiences with the online unit as a precursor to creating the entire course online for the next iteration.

Study2 was conducted in the same classroom with a new cohort and used a version of FACT on OpenEdX with roughly 60 videos (one to five minutes in length), ten quizzes that were interspersed through the course, and learning sequences for blended learning comprising a mix of activities to be done individually or collaboratively. In addition, several refinements were made to the curriculum based on experiences and student feedback in Study1.

Table 1. Student Samples in Study1 & Study2

Study	Mean Age	Count by Gender		Count by Grade		Count in Sp. Programs	
		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

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."

The studies were conducted over two 10-week periods in 2013– March-May (towards the end of the 2012-2013 academic year) and September-November (in the beginning of the 2013-2014 academic year). These 10 weeks included the two weeks before and after the six-week long FACT intervention that included visits to the classroom for IRB permissions, pre-post tests, as well as post-course activities such as final projects and presentations. FACT was taught as part of the semester-long *Computers* elective that met four times a week for 55 minutes each. Students from grades 7 and 8 typically elect to take this course, but approximately a fifth of the class comprised students who had been placed in the class by the school counselors. In both instances, these students happened to be English language learners or students with other learning difficulties who could not be accommodated in other elective classrooms. Unfortunately, since this was an elective class, these students did not get the same specialist supports they received in core subject classes.

Data Measures

The following table summarizes the data that were gathered in the two studies in addition to classroom observation. This difference between studies was due to (a) the online modules of the second course that allowed for additional data to be captured on the online course platform, and (b) lessons from the first iteration of the study were incorporated as improvements in the curriculum and research design as per DBR principles resulting in additional data measures.

Table 10.1. *Main instruments for capturing relevant data measures*

Instrument	Pre-Intervention	Post-Intervention	Source(s)
Computational Knowledge Test	✓	✓	Ericson and McKlin (2012); Meerbaum-Salant, Armoni and Ben-Ari (2010); Zur Bargury, Pârv and Lanzberg (2013)
Preparation for Future Learning (PFL) Test		✓	Designed (Inspired by Schwartz and Martin (2004))
Prior Experience Survey (programming experience and technology fluency)	✓		Adapted from Barron (2004)
CS Perceptions Survey	✓	✓	Ericson and McKlin (2012)
Online Learning Experience Survey (Only Study 2)	✓	✓	Designed (Inspired by depth and breadth of experience items as in Barron (2004))
FACT Experience survey		✓	Designed (for getting student feedback

			for future improvements)
Classroom Observations			During the interventions for DBR

FACT Curriculum

The 6-week FACT curriculum (Table 2) was inspired by the 36-week long Exploring CS high school curriculum (http://www.exploringcs.org/) and included topics that were considered by the authors as foundational and appropriate for middle school students. The entire curriculum design effort was guided by goals for deeper learning, which attend to the development of cognitive abilities through mastery of disciplinary learning for transfer, in addition to interpersonal and intrapersonal abilities.

Table 2. FACT Curriculum Unit-level Breakdown

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)

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 during 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 content and understanding goals of the course, and also to provide feedback both to the learners and the teacher. PFL and transfer of CT skills to text-based computing contexts was mediated through expansive framing (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 as well as programs coded in Scratch. Summative assessments included a specially designed PFL test in addition to other performances of computational understanding.

FACT also consciously engaged with students' narrow perceptions of CS to help them see computing in a new light. For this, publicly available videos were curated that illuminated computing as a creative and problem-solving discipline with applications in many real-world contexts and fields of human endeavor.

Refinements to Learning Environment The improvements to Study 1 that were incorporated in the online/blended version of FACT as part of attempts to refine the curriculum for Study 2 are detailed in Table 3. The key points of distinction between the two studies were:

- Online/Blended versus (mostly) face-to-face.

- Small modifications in content sequencing.
- More time devoted to loops and variables.
- Shorter length of videos (~8-10 mins shortened to ~1-5 mins in Study 2).
- Addition of Scratch window below video for students to be able to try things as they watched.
- Improved video selection for *'Computing is everywhere!'*.
- Improved Scratch Assignments (more games and creative artifacts).
- Online space for contextual questions (below each video).
- An FAQ tab to get answers to online course navigation questions
- A 'Word Wall' tab for computational vocabulary.
- More intentional project-based culminating performance of assessment (Final project, whole-class demo, and online showcase).
- Improved survey question design.

Table 3. *Course Design Changes in Second Iteration of Design-Based Research*

	Study 1	Study 2
Medium of Instruction	Face-to-Face with one online unit	All online on OpenEdx
Duration	6 weeks	6 weeks + 1 week for final project
Student interactions	Restricted to open-ended pair projects and helping others fix Scratch assignments	Additional student interactions in answering thought questions and tackling certain quizzes; in addition to mid-course and culminating project in pairs
'Computing is Everywhere!' video corpus	Included videos that were voted down (such as 'iPhone as a hearing aid' and readings on computation for cancer detection & big data)	More engaging corpus of 'Computing is Everywhere' videos, including Google's Driverless Car; Boston Dynamics' Big Dog robot; IBM Watson and Jeopardy!
'Computing is Everywhere!' video use	Restricted to Week 1 only	In addition to the first week (during Unit 1), one such video was added to the materials for Week 2 to 6, to continue giving learners more examples of cool applications of CS (e.g., Mars Rover singing Happy Birthday to itself; Scratch and Xbox Kinect; and others.)
Videos created	In pilot online unit 8 videos between 4 and 12 minutes (average ~8 mins) in length	60 videos, between 1 and 5 minutes in length
Quizzes	Created on Schoology, Auto-graded	Created on Stanford OpenEdX, Auto-graded, performance data captured in OpenEdX dashboards
FAQ for online course	None	Created a detailed FAQ section that outlined details on navigating through the course e.g., what to do if students had questions, did not understand something,

Assessments	Pretest; Posttest; PFL test	Ongoing quiz scores; Pretest; Posttest; PFL test; Final projects and project presentations; artifact-based student interviews
Sequencing of Content	Pseudo-code & problem-solving paradigm introduced in Week 2	Pseudo-code and problem-solving paradigm introduced in Week 4
Loops & Variables	9 days dedicated to these topics	12 days dedicated to these topics
Boolean Logic	Introduced with Conditionals	Introduced with Repeat Until construct (before Conditionals)
Scratch Examples & Assignments		Used more games, more creative artifacts, more engaging assignment projects (e.g., dropped Assign a Grade, Simple Generic Polygon Maker; added 4-Quadrant Art, Pong)
Vocabulary development	No special effort beyond providing definition while introducing concept	Created a 'word wall' course tab where terms and definitions were added. The teacher also printed this out and pasted on wall just outside classroom door
Final Project	Not a requirement (due to constraints described below)	Elaborate final project requiring students to also document the experience from pre-project planning to post-project reflections
Final Projects Demo	Small informal demos with students who chose to share	Final Projects 'Expo Day' with each group demonstrating their project to the class
Student Interviews	None	In order to assess students' understanding of their programs and thinking behind the code, students were interviewed on the workings of their final project. This was done especially since most students did their projects in pairs. It was also done to compare and contrast students' computational learning as assessed by the posttest.

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 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.

Some gender differences were observed with girls performing better than boys in both studies, however the small number of females in the sample precluded drawing deeper conclusions. 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 ("make/fix/study computer") to a more sophisticated understanding of CS as a problem-solving discipline that uses the computer as a tool to solve real-world problems in many diverse fields.

Comparative Analysis of Study 1 vs. Study 2

How did the students using the blended version of the FACT course (Study 2) perform compared to those in the mainly face-to-face version (Study 1)? This section presents the results of comparative analyses of data from the two studies.

Computational Learning and PFL (Transfer)

On the main outcomes of interest of computational learning, namely the posttest score and the PFL test score, the students in the two studies performed largely the same. There was no significant difference in the posttest and PFL scores. The pretest scores comparing the two studies were also not statistically significant. *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 Study 1 and Study 2 was not significant. These findings are presented in Table 4.

Table 4. Comparison of Main Variables for Computational Learning and PFL, Study 1 vs. Study 2

	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. *Posttest Scores by CT Topics, Study 1 versus Study 2*

	Study 1	Study 2				
Variable	Mean (SD)	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
Vocabulary	68.2 (17.9)	77.4 (22.2)	-1.7	0.10	-2.0	0.05*

On the breakdown of posttest scores by CS topics taught (Table 5), students in both studies performed better on serial execution than on conditionals, which in turn showed better results than loops. Students scored the highest on serial execution, which is the simplest kind of control flow. 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).

Perceptions of Computing

Students in Study 2 appeared to perform better overall than those in Study 1 on the main test of growth of students' perceptions of computing, namely, responses to the question "*In your view, what do computer scientists do?"* As presented in Figure 1, differences between Study 1 and Study 2 in the key coding categories that the responses were coded on were not significant. Since DBR-inspired refinements in the materials that were incorporated in Study 2 based on observations and student feedback from Study 1 aimed at tackling learners' awareness and understanding of computer science and its applicability, 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?"* as described above–students in Study 2 would do better than those in Study 1. Statistical analyses proved this hypothesis to be true: the differences across the two studies (as measured by t-tests and Rank Sum tests) were significant, with students in Study 2 performing significantly better. The difference in the post-course responses as measured by the number of non-naïve codes (richness) was significant at $p<0.01$, and the increase in response length from Study 1 to Study 2 was also significant at $p<0.001$ (Figures 2 and 3, Table 6).

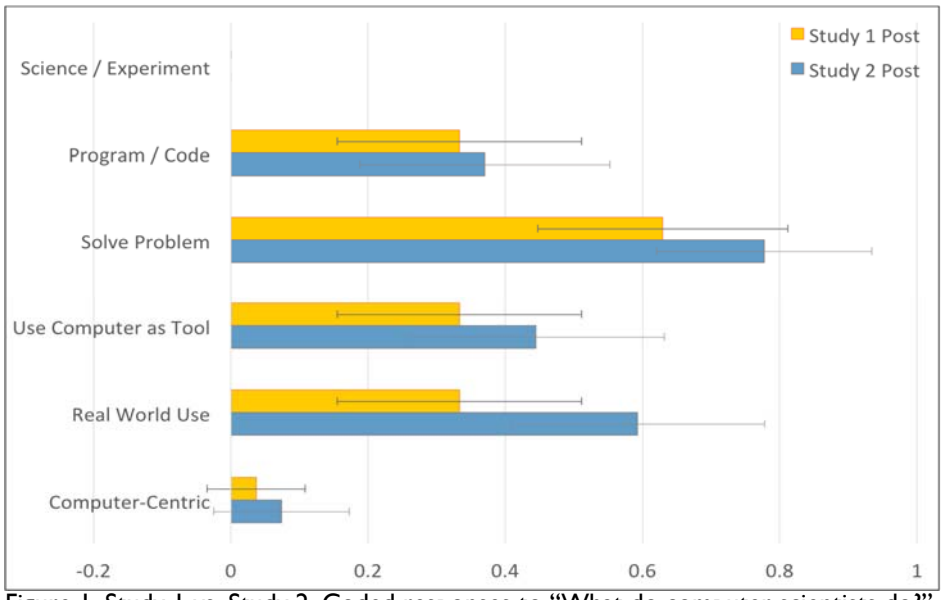

Figure 1. Study 1 vs. Study 2, Coded responses to "What do computer scientists do?"

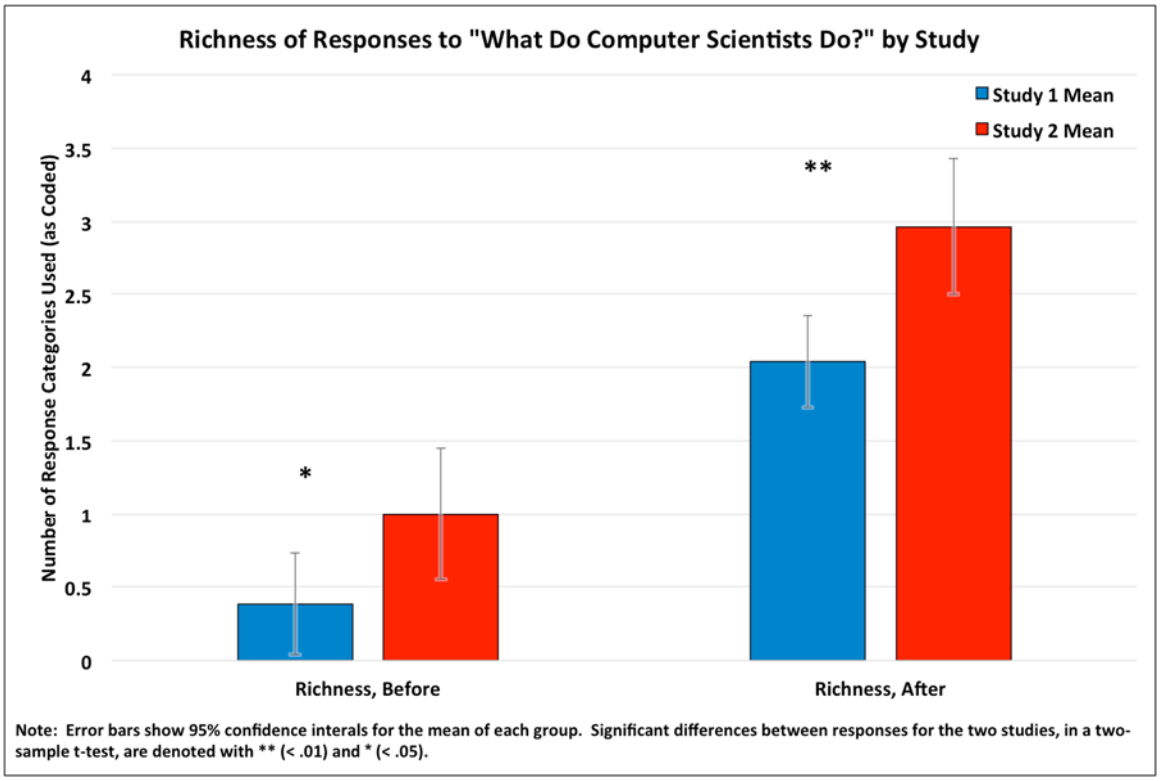

Figure 2. Richness of responses to "What do computer scientists do?", Study 1 vs. Study 2

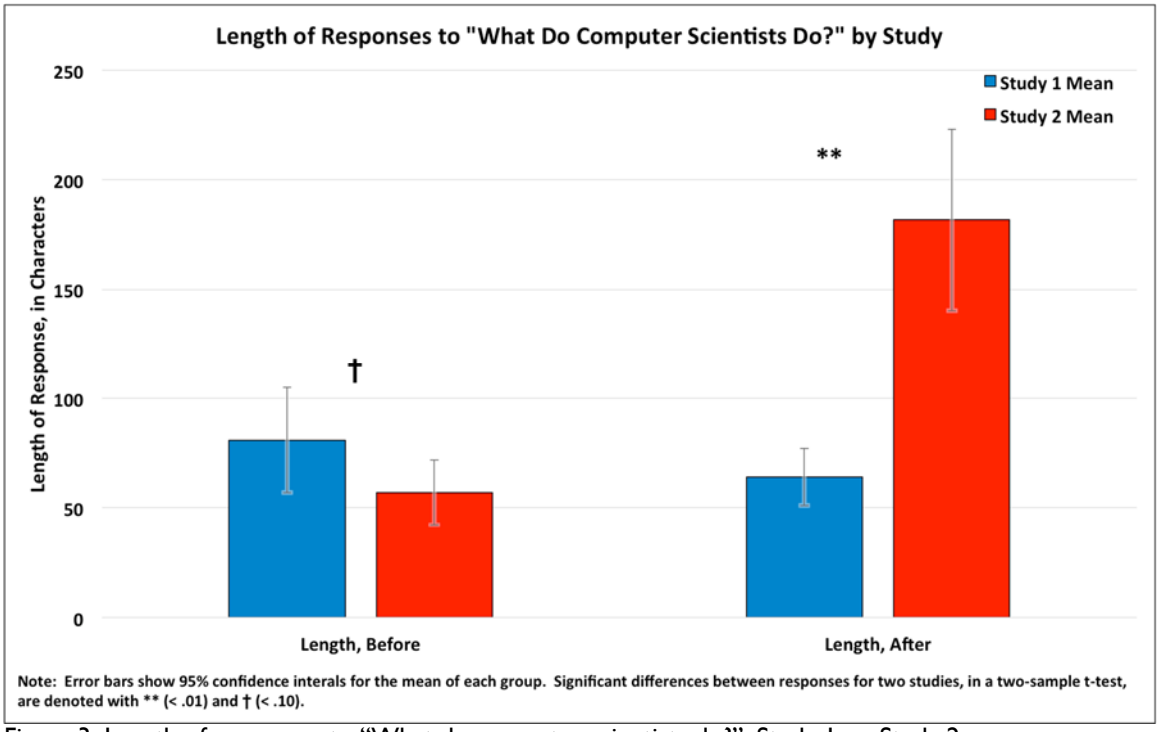

Figure 3. Length of responses to "What do computer scientists do?", Study 1 vs. Study 2

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

Discussion

As these results reveal, Study 2 using the blended version of FACT on OpenEdX worked just as well, if not better, than Study 1–an intervention which used the face-to-face version of the FACT curriculum (except for one pilot unit that was deployed online). This result is even more encouraging taking into account students' preferences for face-to-face learning over online learning (as revealed in pre and post surveys in Study 2). This suggests that the improvements in the FACT curriculum between Study 1 and Study 2 managed to overcome any negative effects or challenges learners may have had with an online/blended course as opposed to a face-to-face one. These results were heartening as they suggested that the iterative refinements driven by DBR resulted in an online FACT course that was a success in a real classroom setting.

It should be noted that the improvements in Study 2 were observed in the sections that had undergone refinements after the first iteration of this design-based research investigation. These refinements included improved strategies for helping learners internalize the vocabulary of computing. They also targeted ways in which students could better appreciate the ideas behind the '*Computing is Everywhere!*' unit of the curriculum. The design-based research effort appears to have resulted in an improved curriculum in these aspects where re-design was enacted.

Final Project, Presentation and Interview as Performance Assessments in Study 2

Study 1 did not have a formal project at the end of the intervention, as the weeks following the intervention were disruptive as they marked the end of the school year and were dedicated almost entirely to end-of-year activities and planning for the 8th grade graduation. However, the success of the final open-ended project for the few children that did do it (informally), prompted a decision to add a 7th week in the curriculum in Study 2. After the six weeks of the FACT curriculum ended, students worked on a culminating project that involved designing and programming a game of their own choosing in Scratch which they then demo-ed to the class on a designated 'Project Expo Day'.

Week 7 was filled with a lot of hard work and enthusiastic coding. Many students made plans to continue working on their projects in the evenings in order to complete them on time. 'Project Expo Day' was marked by much excitement. Every student or student-pair came up to the teacher's computer that was connected to the projector. They demonstrated their projects one after another. Students had some experience with such a process as the mid-course open-ended projects had been similarly shared with the entire class. Sometimes they displayed their code to show how some aspect of the game was programmed. At other times, they called on other students to demonstrate games that required two or three players. The class was appreciative of every project, and every presentation was followed by enthusiastic applause. All the final projects were also uploaded to an online 'studio' on the Scratch website as a project "showcase". This allowed students to see all the projects, and play the games created by their peers. 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 document adapted from the *Starting from Scratch* curriculum (Scott, 2013) that was provided to

each student. Most importantly, students played with each other's games (or their own). In their post-survey responses, students were provided the list of all the projects they worked on in course of the FACT course, and asked to vote the **three** 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 making sense of Scratch code and answering questions based on them, the final project was a more meaningful form of performance assessment for which Barron and Darling-Hammond (2008) argue. It embodied learner agency and appeared to instill a sense of accomplishment in every student as they presented their projects to the class and received cheers of praise from their peers. Most importantly, it seemed to have worked well even for the students who performed poorly on the posttest.

These observations of the final project experience juxtaposed with the posttest performance suggested that perhaps for some students the experience of creating a game was a more rewarding and engaging 'performance assessment' to showcase their learning. Additionally, regression analyses suggested that the text-heavy computational knowledge tests appeared to disadvantage ELL students, a qualitative analysis of final projects and interviews was also conducted for students who scored in the lowest percentile in the posttest. These revealed that students' final projects evinced high levels of engagement and showed evidence of understanding of program construction and algorithmic constructs (Table 7) that the posttest was unable to capture. This represented was one more significant improvement in Study 2 inspired by DBR.

Table 7. Highlights of "Artifact-Based" Interview with Isaac (I) who was among the lower performers on the CT posttest and PFL test

Interpretation/Question posed	***Student Quote(s) or Exchanges with Interviewer (X)***
Connected final project idea with event in personal life	*"it was becoming Halloween so we thought that we should do like a scary maze… This was the only scary maze that I knew about, coz my cousin made me play it"*
"What made you think of the Insidious song?"	*"Ooh. it seems.. it's a creepy little song. It's like .. yeah.."*
Able to describe what the code did, referred to the code elements while talking.	*e.g. we switched it to the uh.. first maze.. and then we put the sprite to the beginning.. and then we did a Forever loop..and we did "If.. the key. up arrow is pressed.. change y by 1;and if the down key is pressed, change y by negative… then if touching black uh.. you re-start it to the beginning, and um if you touch red you um you um change backgrounds to the second maze."*
Knew what the bug was; showed it and had some idea what could fix it without X's help; realized that a variable was needed to keep track of the level and knew where that variable would be set and updated.	*And as you go to the black, it just takes you down.. It's coz we had to like put something in between and .. it switches the backdrop. so like in between one of these things.. or... if./ else I think, you put in there.* *X: where would you change that variable..?* *I: whenever it touches the red..* *X: very good! and then when it touches the black..?* *I: It depends on what level it is, it would go over there..*
Found the project to be fun and creative	*X: So in general if you had to say whether it was fun or challenging or difficult or easy or creative or boring which ones would you choose.* *I: Fun* *X: anything else?* *I: Creative*

	X: What was the creative part of it? *I: Like putting, like, the face at the last, and the screaming..*

Reflections on Study 2

Although the online FACT course designed for blended learning could be considered a success per the criteria guiding this research, based on classroom observations and student feedback, it appears that the course design could have done more to attend to learner agency. Perhaps if there was more freedom and flexibility with respect to the length of the course, students could have been left to work on more guided explorations, and choose from different projects to tackle as part of their Scratch assignments. In a live face-to-face class, projects and examples may have emerged from classroom discussions or out of topics of interest to the students taking the course. These examples were attempted in FACT through the iterative design process by getting students to vote on their favorite assignments and projects in both studies. In fact, learners in Study 1 had suggested that there should be more game projects which prompted a re-design of some of the examples and Scratch exercises for Study 2.

Despite allotting more time in Study 2 for hard-to-learn concepts such as variables, looping and abstraction, learners still appeared to struggle with those ideas, especially those with poor Math preparation. This suggests that perhaps we need to rethink of the way those ideas are introduced to learners and use insights from abstract thinking in disciplines such as Math to help learners.

These results suggested that the PFL test would benefit from redesign. For a balanced set of questions with robust construct validity, the PFL test should test learners on the individual concepts taught–serial execution, variables, conditionals and loops–in addition to some advanced snippets of code that incorporate all concepts rather than only loops and variables as was the case. Additionally, the PFL test should be designed such that it does not disadvantage learners who have difficulty with the English language. The surveys before and after the course as well as some of the questions on the quizzes, and many of the questions on the posttest and PFL test required students to read large amounts of text. This was a challenge for ELL students and learners not very fluent with English. Such students sometimes 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 the English language, either through simpler scenarios leading up to questions, or by providing materials in multiple languages in addition to English, such as Spanish. This would not be too difficult since FACT is an online course, and technology could be leveraged to present materials to learners in the language of choice. The video transcript could also be provided in languages other than English.

Congruent with methodologies of design-based research, 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. Lastly, some students clearly would have benefited from more time as having to stick to the schedule of the research study meant that some students lagged behind in the course. This needs to be improved.

Conclusion and Scholarly Implications

Study2 using the online version of FACT in a blended in-class setting worked just as well, if not better, than Study1, an intervention that used the face-to-face version of the FACT curriculum. Based on success metrics for an online *replacement* course (Means, et al., 2010),

online FACT was successful since learners did at least as well as those in the face-to-face intervention.

This research serves as an example of course design on a MOOC platform that employs DBR to refine a curriculum aimed at fostering deeper learning in a blended classroom setting. 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. Through the use of pedagogies for guided instruction and active learning, FACT is quite different from most other MOOCs of today. 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 (in a manner of speaking) made for a more active learning experience. Incorporating a wide range of mechanisms to learn, hone and demonstrate computational thinking is what make FACT a unique and innovative computer science curriculum. This research also exemplified the value of design-based research (DBR) as a methodology to iterate on the design of the online course. In their role as "design partners", students provided detailed 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 the second iteration and using FACT with broader audiences of middle school students and teachers across the US. As von Hippel and Tyre (1995) and Engeström (2008) rightly point out about the essential nature of design-based research– 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 with further improvements in FACT. FACT currently embodies a well-thought-out, pedagogically-robust, design of a curriculum for computing in middle school. Nonetheless, it still has room for improvement. This research represents only 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

Amiel, T., & Reeves, T. C. (2008). Design-Based Research and Educational Technology: Rethinking Technology and the Research Agenda. *Educational Technology & Society*, *11*(4), 29-40.

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *The 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., & Daring-Hammond, L. (2008). How can we teach for meaningful learning.? In Daring-Hammond, L., Barron, B., Pearson, P. D., Schoenfeld, A. H., Stage, E. K., Zimmerman, T. D., Cervetti, G. N., & Tilson, J. L. *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, A. L., & Campione, J. C. (1994). Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (pp.229-272). Cambridge, MA: MIT Press.

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. *The Journal of the learning sciences*, *13*(1), 15-42.

Engeström, Y. (2009). The future of activity theory: A rough draft. *Learning and expanding with activity theory*, 303-328.

Engle, R. A., Lam, D. P., Meyer, X. S., & Nix, S. E. (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.

Markus, H., & Nurius, P. (1986). Possible selves. *American Psychologist, 41,* 954-969.

McKenney, S., & Reeves, T. C. (2014). Educational design research. In *Handbook of Research on Educational Communications and Technology* (pp. 131-140). Springer New York.

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.

Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M., (2010). Learning computer science concepts with Scratch. In *Proceedings of the Sixth International Workshop on Computing Education Research (ICER '10)*. ACM, New York, 69-76.

Middleton, J., Gorard, S., Taylor, C., & Bannan-Ritland, B. (2008). The "compleat" design experiment: From soup to nuts. *AE Kelly, JY Baek, & RA Lesh, Handbook of design*

research methods in education: Innovations in science, technology, engineering, and mathematics learning and teaching. Routledge, 21-46.
Pea, R. D. (1987). Socializing the knowledge transfer problem. *Int'l Journal of Ed. Research*, 11(6), 639-663.
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.
Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, *13*(2), 137-172.
Sandoval, W. A., & Bell, P. (2004). Design-based research methods for studying learning in context: Introduction. *Educational Psychologist*, *39*(4), 199-201.
Schwartz, D. L. & 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. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard educational review*, *57*(1), 1-23.
Shulman, L. S. (1994). Those who understand: Knowledge growth in teaching. *Teaching and learning in the secondary school*, 125-133.
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.
Wing, J. (2011). Research notebook: Computational thinking—What and why? *The Link Magazine*, Spring. Carnegie Mellon University, Pittsburgh. Retrieved from http://link.cs.cmu.edu/article.php?a=600
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.