Improving Learning of Computational Thinking Using Computational Creativity Exercises in a College CS1 Computer Science Course for Engineers

Duane F. Shell,
Melissa Patterson Hazley
Department of Educational Psychology
University of Nebraska, Lincoln
dshell2@unl.edu,
mpatterson.hazley@gmail.com

Leen-Kiat Soh, L. Dee Miller,
Vlad Chiriacescu
Department of Computer Science and Engineering
University of Nebraska, Lincoln
{lksoh, lmille, vchiriac}@cse.unl.edu

Elizabeth Ingraham
Department of Art and Art History
University of Nebraska, Lincoln
eingraham2@unl.edu

Abstract*—Promoting computational thinking is a priority in CS education and other STEM and non-STEM disciplines. Our innovative, NSF-funded IC2Think project blends computational and creative thinking. In Spring 2013, we deployed Computational Creativity Exercises (CCE) designed to engage creative competencies (Surrounding, Capturing, Challenging and Broadening) in an introductory CS1 course for engineering students. We compared this CCE implementation semester (80 students, 95% completing 3 or 4 CCEs) to the Fall 2013 semester of the same course (55 students) without CCEs. CCE implementation students had significantly higher scores on a CS concepts and skills knowledge test (F(1, 132) = 7.72, p < 01, partial Eta^2 = .055; M=7.47 to M=6.13; 13 items) and significantly higher self-efficacy for applying CS knowledge in their field (F(1, 153) = 12.22, p < .01, partial Eta^2 = .074; M=70.64 to M=61.47; 100-point scale). CCE implementation students had significantly higher study time (t(1, 136) = 2.08, p = .04; M=3.88 to M=3.29; 7-point scale) and significantly lower lack of regulation, which measures difficulties with studying (t(1, 136) = 2.82, p = .006; M=2.80 to M=3.21; 5-point scale). The addition of computational creativity exercises to CS courses may improve computational thinking and learning of CS knowledge and skills.

Keywords—CS1, Creative Thinking, Computational Thinking, Engineering, Student Learning, Self-Regulation, Engagement, Self-Efficacy

I. INTRODUCTION

Promoting computational thinking is a top priority in Computer Science (CS) education [1, 2]. Recent articles in Communications of the ACM show a collective momentum to enhance computational thinking education within CS and across STEM and non-STEM disciplines. This research is diverse, ranging from course specifications to course development, from community building to setting policies, and from teaching and learning to assessment [3, 4, 5, 6, 7].

In our NSF-funded IC2Think project, we proposed that learning computational thinking can be improved through synergy between computational and creative thinking. Our innovative solution was to blend computational thinking with creative thinking. By blending computational and creative thinking students can leverage their creative thinking skills to "unlock" their understanding of computational thinking [8]. In this way, we can make computational thinking more generally applicable to STEM and non-STEM disciplines where students may have creative thinking skills but lack understanding of computing concepts. The reverse is also true: students who understand computational thinking could leverage it to improve their creative thinking skills. We have referred to this blending as *computational creativity*.

In our framework, both computational thinking and creative thinking are viewed as cognitive tools which expand the knowledge and skills that one can apply to a problem. Computational thinking is an approach to solving problems, building systems, and understanding human behavior that draws on the power and limits of computing [1]. Computational thinking includes skills such as conceptualizing at multiple levels of abstraction, defining and clarifying a problem by breaking it down into relational components, and testing and retesting plausible solutions. Creative thinking is thinking patterned in a way that tends to lead to creative results [9]. Creative thinking is not limited to the arts but is an integral component of human intelligence that can be practiced, encouraged and developed within any context [10, 11, 12]. Epstein's Generativity Theory breaks creative thinking down to four core competencies: *capturing* novelty, *challenging* established thinking and behavior patterns, *broadening* one's knowledge beyond one's discipline, and *surrounding* oneself with new social and environmental stimuli [13].

The blending of computational thinking with creative thinking is not conceived as a set of dichotomies, but rather as complementary or symbiotic abilities and approaches. Computational thinking tools expand the knowledge and skills that one has available thereby broadening the array of knowledge that one may creatively apply to a problem. Similarly, creative competencies enhance the development of computational thinking. For example, *Challenging* forces computational tools to be used in unanticipated and unusual ways leading to the development of new computational approaches to both old and new problems, *Surrounding* creates new ways of looking at problems and attention to different stimuli and perspectives that may be relevant to how a problem is approached computationally, and *Capturing* forces consideration of new ways to represent and save data and solution procedures.

We have designed computational creativity exercises (CCE) that blend computational and creative thinking. The principles underlying the design of the CCEs are (1) balancing of attributes between computational and creative thinking and (2) mapping between computational and creative concepts and skills as they manifest in different disciplines. Table 1 shows the differing attributes, concepts, and skills that are balanced and mapped.

TABLE I. MAPPING COMPUTATIONAL AND CREATIVE THINKING SKILLS IN THE CCES

MAPPING CONCEPTS & SKILLS	
Computational Thinking	**Creative Thinking**
Algorithmic Thinking – algorithms as models of computational processes*	Challenging accepted solutions and procedures
Programming Fundamentals - i.e. data models, encapsulation, testing and debugging*	Broadening applications through new ways of framing and applying fundamental computational knowledge
Computing Environments – languages & paradigms, tools, applications*	Surrounding with new social and physical environments to broaden perspectives
Data representation: data types, variables*	Capturing new ways of storing and representing data & algorithms

*CC2001: IEEE/ACM Computing Curriculum, 2001

The CCEs provide instruction on CS concepts such as conditionals, arrays, modular programming, and debugging; however, they completely lack any programming code. Instead, the CCEs involve tasks seemingly unrelated to CS such as individually writing the chapters for a story based on separate plot points and then working in a group to revise the story to make the content more cohesive. This makes students focus on and apply the more abstract computational thinking principles that underlie the CS concepts. CCEs also are designed to foster development of Epstein's creative competencies by engaging multiple senses, requiring imaginative thought, presenting challenging problems and combining both individual and group efforts. During the CCEs, students are given explanations, which we call CS Light Bulbs, which relate the tasks directly back to the CS topics. Finally, the students answer analysis and reflection questions designed to further promote both computational thinking and creative application of computational skills.

II. PRIOR WORK

In the Fall 2012 semester, we designed and deployed four CCEs at the University of Nebraska-Lincoln. Over 200 students in four different introductory CS1 courses took the exercises and the exercises counted as part of their final course grades. Each course was tailored to a different target group (CS majors, engineering majors, combined CS/physical sciences majors, and humanities majors) so the courses contained a mix of students with differing levels of both computational and creative thinking skills. At FIE in 2013, we reported a "dosage" effect of higher grades and learning of core computational thinking principles associated with increasing CCE completion across these courses in Fall 2012 [14]. Also at FIE in 2013, we reported associations between creative competency and course grades supporting our contention of synergy between creative and computational thinking and associations between creative competency and higher strategic self-regulation across the four CS1 implementation courses and one additional CS1 course [15].

At SIGCSE 2014, we reported on follow-up studies of the Fall 2012 semester implementation examining differences between CS majors and non-CS majors, freshmen and upper class students, and men and women students. We found that for all students the linear "dosage effect" was present with student completion of each additional exercise increasing retention. For grades, however, the effects were more nuanced with CS majors having a consistent linear increase for each exercise completed, whereas non-majors had grade increases only for completing at least three exercises. Also, upper class students had increases for completing at least three exercises, whereas freshmen students needed to complete all four exercises. We did not find differences between women and men students but could not draw conclusions because of restricted sample size [16].

Finally, at the 2014 American Educational Association Annual Meeting, we reported on additional follow-up studies of the Fall 2012 semester examining impacts on exercise completion on creative competency and motivational influences on exercise completion. We found that exercise completion did not impact overall creative competency scores or scores for any of the four Epstein creative competencies. We attributed this failure to find an effect primarily to limitations with the Epstein Creative Competency Inventory measure. We found that ccompletion of exercises was motivated by higher task approach goals and perceived instrumentality and lower negative affect [17].

Focus groups and open-ended survey responses [14] have shown that students report not enjoying the exercises and seeing little connection between the exercises and the course content. They express that the exercises did not have enough payoff to justify the time they took. *These findings stand in contrast to the apparently strong impacts of exercises on grades and learning*. When probed further in focus groups for exercise impacts on specific computational thinking and computing skills [14], many students indicated that these were improved by completing the exercises and that exercises helped them understand how programming relates to the world around them and how they might apply concepts in a practical setting. These findings suggest that students are not necessarily able to recognize the value of educational activities for enhancing their learning, especially in their first or general impressions of an activity. We think that this negative impression of the exercises may explain our findings [15] that learning approach goal orientation and Future Time Perspective connectedness were *negatively* associated with completing more exercises. If students didn't see connection between the exercises and learning, those with high learning approach goals might not have been motivated to do the exercises. Similarly, students with more connection to their future career may not be motivated to complete exercises if they do not see them as being relevant to the course and their career.

III. THE PRESENT STUDY

The purpose of this study was to expand on our previous findings by conducting a quasi-experimental trial. The same four Computational Creativity Exercises (CCEs) we previously used were implemented in Spring 2013 in the introductory CS1 course tailored to engineering majors using MATLAB as the programming language with labs on various engineering applications. Based on student feedback from prior implementations, the CCEs were emphasized more in class and more closely aligned with course topics. In the Spring 2013 CCE implementation semester, 95% of students completed either three or four CCEs. Student outcomes from this semester were compared to the Fall 2013 semester of the same course taught without CCEs. The CS1 engineering course has students from a variety of engineering majors. Most students in the two semesters were in chemical engineering (68), civil engineering (54), and mechanical engineering (85).

A. Computational Creativity Exercise Design and Deployment

All CCEs combine hands-on problem solving with written analysis and reflection and are designed to be completed by a group of students working collaboratively in order to facilitate creative thinking from students with multiple backgrounds. CCEs combined Week 1 hands-on problem solving (group and individual) and Week 2 written analysis and reflection that pushed students to consider more effective problem-solving approaches and possible real-world applications. Problems ranged from designing and cracking ciphers to constructing and debugging a coherent story using disconnected plot points. CCEs had embedded self-contained explanations (CS Light Bulbs) that made explicit the connections between exercise tasks and CS topics. The CS Light Bulbs help students stay aware of the underlying computational thinking skills while they are performing the creative thinking tasks. The exercises were deployed using the Written Agora system [18]. This is a web-based wiki system designed to facilitate online collaboration between groups of students, anchored around content pages authored by the students with their corresponding discussion forums, where students contribute their responses to the analysis and reflection questions. The Written Agora system also tracked student interactions with the system online and kept all the page revisions so that we could determine which students were contributing to the group.

Each CCE requires approximately 1-2 hours per student. After completing the tasks and answering the questions, each student group was assigned a base score from which individual grades were further derived base on students' contributions to their group's wiki page. Combined, the exercises represented 5% of the student final grade in the MATLAB course. Additional details on the CCEs are available in previously published work [14, 16].

B. Learning and Self-Efficacy

Our prior work indicates that completion of CCEs improves both students' grades and retention of core computational and CS knowledge [14, 16]. These achievement and learning outcomes are critically important, but research has shown that effective use of knowledge or skill, particularly for transfer and problem-solving, requires self-efficacy or confidence in one's capability of using the knowledge [19, 20]. So, it is important that students not only learn the CS1 course content, but also gain the self-efficacy necessary for applying what they have learned. This is particularly critical for students in required courses that are integral to their major field but not directly a course in the major topic area, like the CS1 course for engineers examined in this study. For these engineering students' to use knowledge and skills learned in this CS1 course, they must gain not only the knowledge but the self-efficacy to apply it.

C. Strategic Self-Regulation and Engagement

We propose that computational creativity can increase not only achievement and learning, but also, students' self-regulation and engagement in courses. Four aspects of student strategic self-regulation have a long history of research.

- The first aspect is general metacognitive self-regulation. Students who are self-regulating are what Pressley *et al.* [21] called good strategy users. These students engage in active planning, monitoring, and evaluation of their learning [22, 23]. In this way, self-regulating students apply general learning strategies to accomplish their goals.
- The second aspect is based on the knowledge building approach proposed by Bereiter and Scardamalia [24, 25]. Students who are knowledge builders go beyond simple factual or recall learning. Learning is tied to personally meaningful goals and includes examination and connection of new knowledge to existing knowledge and coursework in other classes. In this way, knowledge building students are involved with the production of knowledge rather than the reproduction of knowledge.
- The third aspect involves dysfunctional self-regulatory strategies [26, 27, 28] (the opposite of the first aspect). Students who adopt dysfunctional strategies tend to be confused, have difficulty studying and self-regulating effectively, and need support from others. These students are more likely to receive lower grades and have a high likelihood of learned helplessness in classes [26, 27, 28].
- The final aspect is student engagement with the class. Students who are engaged spend more time studying and put forth more effort in class [26, 29, 30]. In particular, students who are engaged are more likely to ask questions than students who are not engaged [30]. In this way, engaged students tend to have more positive experiences in the class and higher achievement [26, 29, 31].

In this paper, we examine whether the CCEs (designed to improve computational and creative thinking skills) have an impact on these four aspects of student strategic self-regulation and engagement. We want to determine whether our contention that completing CCEs boosts students' self-regulation and engagement is accurate by examining the strategic self-regulation of students completing CCEs relative to students in the same course without CCEs.

IV. METHODS

A. Participants

Students voluntarily participated in evaluation data collection which was approved by the UNL Institutional Review Board. In the intervention class, 107 students were initially enrolled of which 13 withdrew. All 107 initially enrolled students (86 male; 21 female) consented to participation and 90 students (72 male; 18 female) completed pre and post-surveys. In the control class, 156 students were initially enrolled of which 23 withdrew. Of initially enrolled students, 120 (100 male; 20 female) consented to participation and 65 students (53 male; 12 female) completed pre and post-surveys.

Because of the high rate of non-completion of post-surveys in the control class, we compared demographic make-up of those who did both pre-and post-surveys and those who did only the pre-survey. There were no differences in gender, year in school, or intention to major/minor in CS. Also, there were no significant differences in any of the pre-survey motivation variables. Students who completed both surveys had significantly higher course grades ($M = 3.36$) than those who did only the pre-survey ($M = 2.51$) indicating that the control class sample for the study analyses using the post-surveys and knowledge test was biased toward higher achieving students relative to the implementation class where almost all students were in the analysis sample.

B. Learning and Self-Efficacy Measures

Students' retention of computational thinking and CS knowledge and skills from the course was measured with a computational thinking knowledge test developed by CSCE faculty [14, 15, 16, 17, 26]. The test contained 13 conceptual and problem-solving questions for the core computational thinking content common to all CS-1 classes. The Cronbach's alpha reliability estimate based on a sample of 391 students from three semesters of the CS1 courses was .74.

Students' self-efficacy was assessed using a questionnaire adapted from [32]. Students were asked to rate their confidence in their knowledge of 12 computational thinking and CS topics with emphasis on application in their chosen field and enhancing creativity in their field ("Your ability to use computational algorithms to solve problems in your field;" "Your ability to conceptualize data in your field in ways that can be analyzed computationally;" "Your ability to think of novel ways of doing things in your field"). Questions were based on the specific knowledge and skills taught and used during the CS1 courses. Ratings were done on a scale from 0 (*Completely Unconfident*) to 100 (*Completely Confident*). Self-efficacy scores were computed as the mean of the 12 items. The Cronbach's alpha reliability estimate was .96.

C. Strategic Self-Regulation Measures

Strategic self-regulation was assessed with three scales from the Student Perceptions of Classroom Knowledge Building instrument (SPOCK) [15, 26, 29, 33]. *Self-regulated strategy use* (5 items) assesses the extent of participant planning, goal setting, monitoring, and evaluation of studying and learning (e.g., "In this class, I tried to monitor my progress when I studied"). *Knowledge building* (5 items) assesses the extent of student exploration and interconnection of knowledge from the class (e.g., "As I studied the topics in this class, I tried to think about how they related to the topics I have studied in other classes"). *Lack of regulation* (4 items) assesses participants' lack of understanding of how to study and need for assistance and guidance in studying (e.g., "In this class, when I got stuck or confused about my work, I needed someone else to figure out what I needed to do").

All questions were answered on a five-point Likert scale from 1 (*almost never*) to 5 (*almost always*). Scores were computed as the mean score of the scale items. Cronbach's alpha reliability estimates for the self-regulated strategy use, knowledge building, and lack of regulation scales were .77, .83, and .78 respectively.

D. Engagement Measures

Engagement was assessed with four measures. Two scales from the SPOCK assess the extent of question asking in class. *High-level question asking* (3 items) assesses the extent to which students ask questions that extend or expand on the basic information being provided in the class (e.g., "In this class, I asked questions to more fully understand the topics we were learning"). *Low-level question asking* (3 items) assesses the extent to which students ask questions to obtain or clarify basic course information ("In this class, I asked questions to be clear about what the instructor wanted me to learn"). Scores are computed as the mean of the items in each scale. Cronbach's alpha reliability estimates for high-level and low-level question asking were .78 and .83 respectively.

Two scales assessed self-reported studying [26, 29]. Study time was assessed by asking participants to indicate the average number of hours per week they spent studying for the class on a 1–7 scale in two hour increments from 1 (<2 h per week) to 7 (over 12 h per week). Perceived study effort was assessed by asking participants to indicate their perception of the effort they put forth studying for the class relative to most students on a 5-point Likert scale from 1 (*much less effort*) to 5 (*much more effort*).

E. Motivation and Course Perception Measures

Assessments of students' motivation and perceptions of the classroom environment were used as covariates along with strategic self-regulatory and engagement measures to control for individual differences in these when testing for impact on learning. Students' motivation was assessed with a battery of instruments which have been validated in prior studies. These consisted of (1) students goal orientation for the class [15, 26, 29]; (2) students' ratings of the connectedness between their academic coursework and a STEM career and instrumentality of their specific course work for attaining STEM academic and career goals [26, 29, 34, 35]; (3) students' emotional/affective reactions to the course [26, 29, 36]; and (4) students' beliefs about intelligence [15, 37]. Students' classroom perceptions were assessed with the collaborative learning and teacher directedness scales from the Student Perceptions of Classroom Knowledge Building instrument (SPOCK) [33].

F. *Procedures*

All measures and the computational thinking test were administered on a Web platform (Survey Monkey®) administered during course lab periods. The pre-test measures were done in labs during the first week of class and the post-test for all survey measures and the knowledge test were done in lab sections during the last week of classes as part of broader evaluation data collection.

V. RESULTS

A. *Impacts on Student Learning and Self-Efficacy*

We used Analysis of Variance (ANOVA) to test whether knowledge test scores differed between the implementation and control semesters (Table II). Students in the implementation semester had significantly higher knowledge test scores ($F(1, 132) = 7.72$, $p < .01$, partial $\text{Eta}^2 = .055$). We used Analysis of Covariance (ANCOVA) to test whether knowledge test scores differed between the implementation and control semesters when controlling for students' course grades (Z-score standardized within class), strategic self-regulation, engagement, motivation, and classroom perceptions. With these potential individual difference variables controlled (Table II), students in the implementation semester still had significantly higher knowledge test scores ($F(1, 106) = 12.78$, $p < .01$, partial $\text{Eta}^2 = .108$). With potential individual difference controlled, the knowledge test score difference and effect size increased.

We used Analysis of Variance (ANOVA) to test whether students' self-efficacy scores differed between the implementation and control semesters (Table II). Students in the implementation semester had significantly higher self-efficacy scores ($F(1, 153) = 12.22$, $p < .01$, partial $\text{Eta}^2 = .074$).

B. *Impacts on Students' Strategic Self-Regulation and Engagement*

We used independent samples (*t*-tests) to examine differences in strategic self-regulation and engagement between students in the implementation and control semesters. Results are shown in Table III. Although students in the implementation semester had higher scores for self-regulation and knowledge building differences were not significant. However, students in the implementation semester had significantly *lower* lack of regulation scores with a moderate effect size (Cohen's $d = .50$). For engagement measures only study time differed. Students in the implementation semester reported significantly more study time with a small effect size (Cohen's $d = .37$).

TABLE II. KNOWLEDGE AND SELF-EFFICACY MEAN SCORES

	Implementation Semester		***Control Semester***	
	M	***SD***	***M***	***SD***
Knowledge Test[a]	7.47	2.56	6.13	3.01
Knowledge Test Controlled[a]	7.47	2.56	5.94	2.93
Self-Efficacy[a]	70.64	14.31	61.47	18.32

[a]All differences significant at $p < .01$.

TABLE III. SELF-REGULATION AND ENGAGEMENT MEAN SCORES

	Implementation Semester		***Control Semester***		
Variable	***M***	***SD***	***M***	***SD***	***t***
SPOCK Self-Regulation	3.22	.66	3.06	.72	1.38
SPOCK Knowledge Building	2.82	.69	2.69	.83	0.94
SPOCK Lack of Regulation	2.80	.85	3.21	.80	2.82**
SPOCK High-Level Question Asking	2.35	.73	2.54	.92	1.34
SPOCK Low-Level Question Asking	2.53	.84	2.65	.83	0.79
Study time	3.88	1.72	3.29	1.48	2.08*
Study effort	3.24	.92	3.28	.97	0.24

*$p < .05$, **$p < .01$.

VI. DISCUSSION AND CONCLUSIONS

Results confirmed our prior findings [14, 16] that the Creative Competency Exercises (CCE) positively impact student learning of computational thinking and CS knowledge and skills. The study strengthened these previous findings by employing a stronger quasi-experimental design with a comparable control group. The study also extended prior work by examining self-efficacy and students' strategic self-regulation and engagement.

Our prior studies [14, 16], examined a broad spectrum of students from a diverse suite of tailored CS1 courses. This study focused specifically on engineering majors in one of the targeted CS1 courses. Most engineering programs require students to take foundational technology, science, and mathematics courses, like the CS1 course in this study, early in their studies. Typically taught outside of the engineering program in other schools or departments, these courses provide necessary scaffolding towards subsequent technical coursework within the students' engineering field. These courses have been referred to as barrier courses because poor performance in them can act as a barrier to continued pursuit of engineering [38]. Because these courses are critical to student success and retention, our findings have particular relevance. Increasing engineering students' success in courses like CS1 may help address the higher failure rates and attrition experienced by engineering and other STEM students [39].

A. *Learning and Self-Efficacy*

The findings provide additional support for our contention that learning computational thinking can be improved through synergy between computational and creative thinking and that creative thinking skills can help students "unlock" their understanding of computational thinking [14, 16]. Engineering students' who completed the CCE's learned and retained more of the core computational thinking and CS course content. *Differences between the CCE implementation semester and the control semester were not only statistically significant but were also meaningful in the context of the real world classroom.* The effect size was almost half a standard deviation (Cohen's $d = .48$) and over a half a standard deviation (Cohen's d = .56) when course grades and individual differences in motivation, strategic self-regulation, engagement, and perceptions of the class were controlled. This suggests real improvement in learning and retention. The effects are even more impressive since the control group was biased toward better, higher achieving students.

It is critical that students be able to utilize what they are learning in required courses that are integral to their major field in their subsequent studies. To use the knowledge and skills learned in required, non-major courses, like the computational thinking and CS knowledge and skills learned in the CS1 course examined in this study, students must feel self-efficacious in their capability of applying this knowledge and skill to their field [19, 20]. Our results indicate that engineering students completing the CCEs have significantly higher self-efficacy for applying computational thinking and CS knowledge and skill in their field. As with learning, the effect size was over half a standard deviation (Cohen's d = .56).

Because our self-efficacy measure is anchored in our notion of computational creativity and emphasizes application and creative expansion of thinking within the students' major field, in this case, engineering, our findings provide additional support for the importance and effectiveness of blending of computational and creative thinking in STEM classrooms. Engineering students completing the CCEs appear to be more confident in their ability to use computational thinking and CS content in their major field. They also appear to be more confident in their ability to leverage computational thinking and CS content to improve their creative thinking within their major field. Further study is needed to see if this increased self-efficacy translates into more creative and effective problem solving by students in their subsequent engineering courses and whether this higher confidence translates into better student retention

B. *Self-Regulation and Engagement*

The findings somewhat supported our contention that computational creativity can enhance students' strategic self-regulation and engagement in courses. Engineering students in the CCE implementation semester reported significantly higher study time suggesting more engagement in the course. Although they did not significantly increase in general strategic self-regulation or knowledge building, they did exhibit significantly lower lack of regulation. Prior research has found that lack of regulation is strongly associated with poor achievement and negative reactions to the course as well as being a major component of both apathy and learned helplessness [26, 29]. The engineering students in the CCE implementation semester were less likely to report the difficulties associated with lack of regulation suggesting that the CCEs may have helped them succeed. As self-direction and student-centered approaches to the classroom are being increasingly applied, our findings suggest that computational creativity and the CCEs may help improve these aspects of the learner centered classroom.

C. *Next Steps*

When combined with our prior work, the findings strength our belief that the merger of computational and creative thinking can improve student learning both of computational thinking and CS knowledge and skills and of creative ways to apply them, even for engineering students who are non-CS majors. We have found positive effects for the CCEs based on Epstein's creative competencies in two separate study populations using both correlational and quasi-experimental designs. In both studies effects were robust with meaningful effect sizes.

We have now expanded our suite of CCEs and are working on employing them in a broader array of both beginning and upper level courses in CS and in other disciplines. We are also combining the CCEs into a stand-alone distance course. Exercises also have been adapted for K-12 and are available on the Google Exploring Computational Thinking web-site. This allows computational creativity to be introduced to younger students, which we hope will help these students better succeed in mastering CS and computational thinking concepts and increase student interest in pursuing CS education at the post-secondary level.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under grant no. 1122956 and in part by a UNL Phase II Pathways to Interdisciplinary Research Centers grant.

REFERENCES

[1] J. Wing, "Computational Thinking," in Communications of the ACM, vol. 49, 2006, pp. 33-35.

[2] J. Wing, "Computational Thinking: What and Why?" in Link Magazine, 2010.

[3] P. J. Denning, "Computing is a Natural Science," in Communications of the ACM, vol. 49, 2007, pp. 33-35.

[4] M. Guzdial, "Paving the Way for Computational Thinking," in Communications of the ACM, vol. 51, 2008, pp. 25-27.

[5] P. J. Denning, "The Profession of IT beyond Computational Thinking," in Communications of the ACM, vol. 52, 2009, pp. 28-30.

[6] G. H. L. Fletcher, "Human Computing Skills: Rethinking the K-12 Experience," in Communications of the ACM, vol. 52, 2009, pp. 23-25.

[7] C. Lewis, M. H. Jackson, and W. M. Waite, "Student and Faculty Attitudes and Beliefs about Computer Science," in Communications of the ACM, vol. 53, 2010, pp. 78-85.

[8] D. F. Shell, D. W. Brooks, G. Trainin, K. Wilson, D. F. Kauffman, and L. Herr, The Unified Learning Model: How Motivational, Cognitive, And Neurobiological Sciences Inform Best Teaching Practices. Netherlands: Springer, 2010.

[9] D. N. Perkins, "Creativity by Design," Educational Leadership, vol. 42, 1984, pp. 18-25.

[10] U. Kraft, "Unleashing Creativity," Scientific American Mind, vol. 16, 2005, pp. 19-21.

[11] M. Michalko, "Cracking Creativity," Ten Speed Press, 2001.

[12] T. Tharp, "The Creative Habit: Learn it and Use it for Life," Simon & Schuster, 2005.

[13] R. Epstein, S. Schmidt, and R. Warfel, "Measuring and Training Creativity Competencies: Validation of a New Test," in Creativity Research Journal, vol. 20, 2008, pp. 7-12.

[14] L.D. Miller, L-K. Soh, V. Chiriacescu, E, Ingraham, D.F. Shell, S. Ramsay, and M. P. Hazley, "Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses," 43rd ASEE/IEEE Frontiers in Education Conference Proceedings, 2013, pp. 1426-1432.

[15] D. F. Shell, M. P. Hazley L-K. Soh, E, Ingraham, and S. Ramsay, "Associations of students' creativity, motivation, and self-regulation with learning and achievement in college computer science courses,"

43rd ASEE/IEEE Frontiers in Education Conference Proceedings, 2013, pp. 1637-1643.

[16] L.D. Miller, L-K. Soh, V. Chiriacescu, E, Ingraham, D.F. Shell, S. Ramsay, and M. P. Hazley, "Integrating computational and creative thinking to improve learning and performance in CS1," 45th ACM Technical Symposium on Computer Science Education (SIGCSE'2014) Proceedings, 2014, pp. 475-480.

[17] D. F. Shell, M. P. Hazley, L.-K. Soh, E. Ingraham, and S. Ramsay, "Impact of creative competency exercises in college computer science courses on students' creativity and learning," Unpublished paper presented at the American Educational Research Association Annual Meeting, Philadelphia, PA, April, 2014.

[18] A. Eck, L-K. Soh, and C. Brassil, "Supporting active wiki-based collaboration," Tenth International Conference on Computer Supported Collaborative Learning, 2013, pp. 176-183.

[19] A. Bandura, Self-efficacy: The exercise of control. New York, NY, US: W. H. Freeman/Times Books/ Henry Holt & Co., 1997.

[20] S. Bergin, R. Reilly, and D. Traynor, "Examining the Role of Self-Regulated Learning on Introductory Programming Performance," in ICER, 2005, pp. 81-86.

[21] M. Pressley, J. G. Borkowski, and W. Schneider, "Cognitive strategies: good strategy users coordinate metacognition and knowledge." in Annals of Child Development, vol. 5, R. Vasta and G. Whitehurst, Eds. Greenwich, CT: JAI Press, 1987, pp. 89-129.

[22] M. Boekaerts and E. Cascallar, "How far have we moved toward the integration of theory and practice in self-regulation?" Educ. Psychol. Rev., vol. 18, pp. 199–210, 2006.

[23] C.E. Weinstein, J. Husman, and D.R. Dierking, "Interventions with a focus on learning strategies," in The Handbook of Self-Regulation, M. Boekaerts, P. R. Pintrich, and M. Zeidner, Eds. San Diego: Academic Press, 2000, pp. 727-747.

[24] M. Scardamalia and C. Bereiter, "Knowledge building." in Encyclopedia of Education, J.W. Guthrie, Ed. New York, NY: Macmillan Reference, 2003, pp. 1370–1373.

[25] M. Scardamalia and C. Bereiter, "Knowledge building: theory, pedagogy, and technology." in The Cambridge Handbook of the Learning Sciences, R.K. Sawyer, Ed. New York, NY: Cambridge University Press, 2006, pp. 97–118.

[26] D. F. Shell and L-K. Soh, "Profiles of motivated self-regulation in college computer science courses: Differences in major versus required non-major courses," J. of Sci. Educ. and Tech., vol. 22, pp. 899-913, 2013.

[27] J. D. Vermunt and Y. J. Vermetten, "Patterns in student learning: relationships between learning strategies, conceptions of learning, and learning orientations," Educ. Psychol. Rev., vol. 16, pp. 359-384, 2004.

[28] C. A. Wolters, "Understanding procrastination from a self-regulated learning perspective," J. of Educ. Psychol., vol. 95, pp. 179-187, 2003.

[29] D. F. Shell and J. Husman, "Control, motivation, affect, and strategic self-regulation in the college classroom: a multidimensional phenomenon," J. of Educ. Psychol., vol. 100, 2008, pp. 443-459.

[30] M. Scardamalia, and C. Bereiter, "Text-based and knowledge-based questioning by children," Cog. and Instr., vol. 9, 1992, pp. 177-1992.

[31] E. A. Linnenbrink, "The role of affect in student learning: a multi-dimensional approach to considering the interaction of affect, motivation, and engagement." in Emotion in Education, P. Schutz and R. Pekrun, Eds. San Diego, CA: Academic Press, 2007, pp. 107-124.

[32] D. F. Shell, G. R. Snow, D. R. Claes, "The Cosmic Ray Observatory Project (CROP): Results of a summer high-school student, teacher, university scientist partnership using a capstone research experience," in Journal of Science Education and Technology, vol. 20, 2011, pp. 161-177.

[33] D. F. Shell, J. Husman, J. E. Turner, D. M. Cliffel, I. Nath, and N. Sweany, "The impact of computer supported collaborative learning communities on high school students' knowledge building, strategic learning, and perceptions of the classroom," J. of Educ. Comput. Res., vol. 33, no. 3, 2005, pp. 327-349.

[34] J. Husman and D. F. Shell, "Beliefs and perceptions about the future: a measurement of future time perspective," Learn. and Individ. Differ., vol. 18, no. 2, 2008, pp. 166-175.

[35] J. Husman, W. P. Derryberry, H. M. Crowson, and R. Lomax, "Instrumentality, task value, and intrinsic motivation: making sense of their independent interdependence," Contemp. Educ. Psychol., vol. 29, 2004, pp. 63-76.

[36] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: the PANAS Scales," J. of Per. and Soc. Psychol., vol. 54, 1998, pp. 1063-1070.

[37] D. S. Yeager, and C. S. Dweck, "Mindsets that promote resilience: when students believe that personal characteristics can be developed," Educ. Psychol., vol. 47, pp. 302-314, 2012.

[38] R. Suresh, "The relationship between barrier courses and persistence in engineering," J. Coll. Stud. Retention: Res., Theory & Practice, vol. 8, 2007, pp. 215-239.

[39] J. J. Kuenzi, C. M. Matthews, and B. F. Mangan, "Science, technology, engineering, and mathematics (STEM) education issues and legislative options," Washington, DC: Congressional Research Service, Library of Congress, 2006.