WIP: DEPICT for Out-Of-School Time (DEPICT4OST): Guiding Undergraduates in the Development and Implementation of Computational-Infused Writing Activities

Ruth Torres
Department of Computer Science
New Mexico State University
Las Cruces, NM, USA
rutorres@nmsu.edu

Sarah T. Hug
Colorado Evaluation & Research Consulting
University of Colorado at Boulder
Boulder, CO, USA
sara.hug@colorado.edu

Adan Delval
Department of Computer Science
New Mexico State University
Las Cruces, NM, USA
adande@nmsu.edu

Wendy Chi
ABC Research & Evaluation
University of Colorado at Boulder
Colorado, BO, USA
chi@colorado.edu

Enrico Pontelli
Department of Computer Science
New Mexico State University
Las Cruces, NM, USA
epontell@nmsu.edu

***Abstract*—This innovative practice WIP paper describes how undergraduate students developed their curriculum development skills while designing an after-school program for learning computational thinking through writing. DEPICT (Discovering Computational Thinking through Creative Writing) is a pioneering project aimed at integrating computational thinking (CT) principles into high school (HS) creative writing (CW) classes. Developed by computer science (CS) undergraduate students, DEPICT for Out for School Time (DEPICT4OST) introduces a set of 16 innovative lessons designed to foster critical thinking, problem-solving, and algorithmic reasoning through the medium of CW. This WIP presents an exploration of the undergraduate experience in apprenticeship in the K12 outreach instructional space, crafting the DEPICT curriculum and the transformative journey of teaching others through the creation of engaging hands-on activities. Data was collected through pre- and post program surveys, classroom observations, and student work samples to assess the impact of the DEPICT program. The analysis involved comparing quantitative survey results, coding qualitative data from observations and students reflections, and evaluating changes in student work to determine the effectiveness of the program in achieving its educational goals.**

Drawing upon their expertise in CS and passion for creative expression, the undergraduate team embarked on the task of bridging the seemingly disparate domains of CT and CW. Through collaborative brainstorming sessions, iterative design processes, and pedagogical research, they developed a diverse range of lesson plans that seamlessly integrate computational concepts into writing prompts, narrative structures, and storytelling techniques. The practices of DEPICT4OST build upon a departmental emphasis on student ownership of outreach activity, which will be described in the paper.

The paper explores the challenges of adapting complex computational concepts for HS students and aligning activities with educational standards. It also highlights the creative solutions the undergraduate team developed to make these abstract principles accessible and engaging through interactive writing exercises, group discussions, and multimedia presentations.

Furthermore, the paper explores the pedagogical insights gained through the hands-on experience of teaching the DEPICT curriculum to HS students. The undergraduates reflect on their role as facilitators of learning, the impact of active learning techniques on student engagement, and the evolution of their teaching practices. By sharing anecdotes, student feedback, and reflections, the paper offers valuable insights into the effectiveness of infusing CT into CW education.

In conclusion, DEPICT represents a pioneering initiative that harnesses the interactions between CT and creative expression to enrich HS education. The paper not only documents the development of the DEPICT curriculum but also underscores the transformative potential of collaborative, interdisciplinary approaches to education. Through the exploration of the undergraduate experience, it offers a roadmap for future endeavors seeking to innovate in the realm of STEAM education.

***Index Terms*—K12 outreach, computational thinking, student apprenticeship**

I. Introduction

Computational Thinking mirrors the problem-solving process employed by computer programmers during the creation of computer programs and algorithms. It promotes a methodical approach to addressing challenges and encourages individuals to formulate solutions in comprehensible terms, suitable for execution by either a computer or another person [1]–[5]. In terms of real-world applications, beyond the scope of CS, CT stands as a potent instrument, fostering the development of effective problem-solving strategies applicable to academic pursuits and everyday life [6]–[8]. In the context of our intricate digital landscape, the principles of CT equip individuals

Identify applicable funding agency here. If none, delete this.

to effectively engage with a wide spectrum of challenges. Consequently, the integration of CT into educational curricula extends beyond the boundaries of traditional CS instruction [9], [10]. DEPICT (**D**iscov**E**r com**P**utational th**I**nking through **C**rea**T**ive writing) investigates a novel intervention to broaden participation in computing of women and Hispanic high school students [11]. The novelty of DEPICT lies in two aspects. First, DEPICT builds on an area that has not been deeply exploited in the past, i.e., the combination of CW and the production of movies/plays. The appeal of this area comes from its popularity among students, especially women, and Hispanics, and the high level of self-efficacy demonstrated by students in the corresponding courses. The second novelty comes from using this creative domain as a target for the infusion of CT - i.e., we use CT as a methodology to teach the creative domain, contending that relevant concepts of CT are already present in the domain, and they simply need to be extrapolated. This allows us to progressively build CT competency through learning the creative domain, leveraging the students' self-efficacy in the creative domain to build self-efficacy in CT.

In this document, we present the conception of DEPICT4OST (DEPICT for Out of School Time) curriculum developed by CS undergrad students from the DEPICT intervention. This work contributes in several ways. First, it strives to enhance the current understanding of effective strategies for facilitating CT-focused learning, outlining an innovative approach to integrating computational thinking into a non-STEM curriculum. Second, it describes the way we conceived and applied this approach, drawing insights from the active participation and feedback of the CS peers and After-School (A-S) instructors. Lastly, we share some noteworthy challenges and preliminary insights gleaned from the experiences of CS students and the potential of CT to engage learners with minimal prior exposure to computing, fostering collaborative problem-solving skills. We anticipate that this document will serve as a valuable resource for fellow researchers and practitioners seeking guidance in designing and implementing similar learning endeavors.

II. Background and Significance

Computing and K12 Education: Code.org [12] reveals a staggering figure of over 400,000 vacant computing positions, along with roughly 71,000 CS graduates entering the workforce. The scarcity of students pursuing CS can be attributed to various factors, among which a crucial one is the lack of CS education in the K12 system. While several other nations have strengthened their efforts in computing education [13], the CS Teachers Association (CSTA) points out a neglect of CS in American high schools. Merely 47% of high schools nationwide offer any CS curriculum, with only 19 states mandating CS education at the high school level [14], [15]. In the state where this study is taking place, only 32% of the schools offer CS content, with only 21% offering AP CS courses; in 2020, only 371 students took an AP CS test, with less than 25% being female. Such lack of high school preparation in CS is a threat to the number of students pursuing CS college degrees; exposure to computing prior to college is a critical motivation towards the pursuit of CS degrees [16].

Research on outreach experiences shows that sharing knowledge with other audiences can improve computing confidence and lead to greater sense of identity in the field [17]–[19]. By working in DEPICT, CS undergraduate students were "near-peers," in that they are less removed from the high school student population served by DEPICT. Undergraduate students had unique opportunities to connect computing to their undergraduate-level perspectives of the field. Undergraduate students involved in DEPICT were also exposed to and encouraged to consider careers in educational settings, or continuation into graduate school. Even if undergraduates do not enter graduates studies, their experience in DEPICT will help them appreciate the educational needs in their communities and help them develop skills they can use in volunteer and service work in their lifetimes.

In this work, we embrace the perspective of Computational Thinking (CT) presented by Cuny and colleagues [20]: "The thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information processing agent". The idea of identifying the core concepts that underlie computing to guide pedagogical efforts [9], [21] and using CT as a general problem-solving skill that has been explored by various researchers, e.g., [1], [10], [22], [23].

Previous research [18], [24] shows that when college students help leading the learning they increased their content knowledge, understood different learning styles, and gained confidence and patience. When undergrad students lead the learning experiences they tend to move from teacher-centered to learner-centered styles of teaching. In other cases undergraduate leaders improved their mastery of course content, enhanced their communication and leadership skills, and developed professionally [25]. Therefore, having a leading and facilitator role can be beneficial to undergraduate students in different ways.

III. Methodology

A. DEPICT Intervention

The initial DEPICT implementation consisted of three phases: (a) Curricular Phase: this phase comprises the actual infusion of CT into the CW curricula for two different courses. (b) Reinforcement Phase: this phase comprises extra-curricular project-oriented activities, aimed at transitioning from infused CT towards more explicit views of CT and developing collaborative and teamwork skills; (c) Extrapolation Phase: this phase is a summer intervention that completes the transition from the infused view of CT to a full-blown preparation for possible STEM pathway. Findings in this document focus on **the reinforcement** phase because CS fellows[1] took most of the active part on it, but the curriculum builds upon the previous phases.

[1]In this paper, we use the terms "students", "peers", and "fellows" interchangeably

B. The CS Fellows from DEPICT

During the semester of Fall 2023, four CS students (2 females and 2 males) embarked on the task of developing 16 different lessons (4 each) that combine topics from CT and Performance and Creative Writing (PCW). The four of them were juniors at the time of the development. Three of them are majoring in CS with minors in Cybersecurity, Software Development and AI. The fourth one is majoring in Creative Media and minoring on CS. The four CS students that develop the DEPICT4OST curriculum joined DEPICT at different stages of the project. Only one female was part of DEPICT at the beginning of the project, the rest joined DEPICT during the second summer camp.

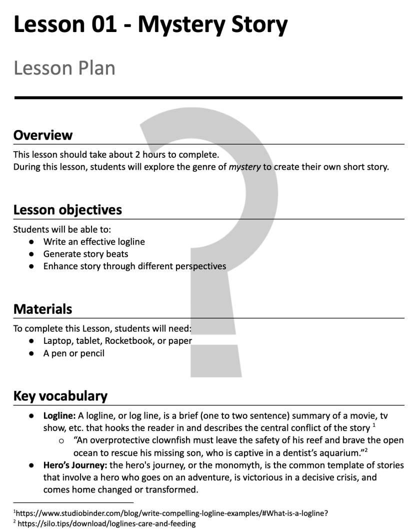
C. How the Curriculum Started

1) School Year Intervention - Curricular Phase: During the academic year 2022-2023, DEPICT embarked on the curricular phase at two local high schools. The CT exposure process included modifying lectures and class activities to emphasize the use of CT concepts to explain CW content. The development of the curriculum for this phase was a collaboration between the HS teachers, the CS students, and the implementation team. We mapped the CW lessons to the CT Big Ideas Guidelines [26] and developed a total of twelve short lessons that included interactive activities and explored the use of technology. See Figure 1.

CW topic	CT/CS related topic
Zen Koans (paradoxical puzzle)	Propositional Logic
Concrete nouns/Connotation/Denotation	Variables
Alliteration (repetition of the same consonant sound at the beginning of the word) *Consonance* (repetitions of the same consonant sound anywhere in the word)	Pattern Recognition and Loops
Scansion: describing the rhythms of poetry by dividing the lines into feet, marking the location of stressed and unstressed syllables, and counting the syllables	Pattern Recognition and Abstraction
Recipe Poem	Decomposition, Pattern Recognition, Algorithmic Thinking
Write a story about an event. Use "But", "yet", "however" to switch the second part of the story	Algorithmic Thinking and If-Else conditionals
Haiku (3 lines that don't rhyme, it has 5, 7 and 5 syllables on each line respectively) Black Out Poetry	Pattern Recognition and Abstraction
Characterization: To do list/Shopping List. Think of a real or fiction character and then make a to-do list or a shopping list for it	Decomposition
Original *Cliché*. Put a new view on something that is familiar. Find new ways to present things	Pattern Recognition
After students are given free writing time, they are asked to write the writing prompt that when followed will create a similar writing piece to the one they wrote	Algorithmic Thinking and Code Analysis
Diagram Poem Can the positions of the words and the path of the lines have some sort of meaning? Can you make things circle around? Can you allow a choice between two paths?	Algorithmic Thinking

Fig. 1. Creative Writing lessons developed during the curricular phase

2) Summer Camp - Extrapolation Phase: DEPICT hosted two two-week camps where local HS students participated in an experience centered around PCW, with a project-focused emphasis on the second week. Throughout the camps, we delved into the worlds of CS, CT, CW, Creative Media (CM), and animation content, using tools like Merge Cubes, Specdrums, Rocketbooks, and Finch Robots. We introduced new software such as PhotoPea and Capcut, web-based applications similar to Adobe Photoshop and Adobe Premiere respectively. Other web-browser applications that were also introduced were CoSpaces, Studio Binder, and Teachable Machine from Google. During these summer intensives, students learned how to bring CS, CT, CM, and PCW together, developing skills to create remarkable projects. A mystery concept was applied to each exercise which aided in collaboration between CS, CM, and PCW. Students applied combinations of traditional and digital techniques to solve riddles, generate whodunit stories, and create imagery intended to deceive their audience and leave them in suspense. Engaging, hands-on activities which gave students a sense of ownership proved to be the most successful. An preview of a summer lesson plan can be seen in Figure 2.


Lesson 01 - Mystery Story

Lesson Plan

Overview

This lesson should take about 2 hours to complete.
During this lesson, students will explore the genre of *mystery* to create their own short story.

Lesson objectives

Students will be able to:
- Write an effective logline
- Generate story beats
- Enhance story through different perspectives

Materials

To complete this Lesson, students will need:
- Laptop, tablet, Rocketbook, or paper
- A pen or pencil

Key vocabulary

- **Logline:** A logline, or log line, is a brief (one to two sentence) summary of a movie, tv show, etc. that hooks the reader in and describes the central conflict of the story [1]
  - "An overprotective clownfish must leave the safety of his reef and brave the open ocean to rescue his missing son, who is captive in a dentist's aquarium."[2]
- **Hero's Journey:** the hero's journey, or the monomyth, is the common template of stories that involve a hero who goes on an adventure, is victorious in a decisive crisis, and comes home changed or transformed.

[1] https://www.studiobinder.com/blog/write-compelling-logline-examples/#What-is-a-logline?
[2] https://silo.tips/download/loglines-care-and-feeding

Fig. 2. Lesson plan that was used during the summer camp

D. Before the Design Process

CS fellows took active part in different strategies that help them build the concepts targeted in the DEPICT4OST curriculum

1) Classroom visits for two consecutive semesters: Twice a week CS fellows implemented different hands-on activities related to the CT concepts while HS students were learning about the host discipline. The HS teacher would then introduce a CW concept that matched the CT concept explored at the beginning of the class. The implementation team documented all the interactions with pictures, field notes, and recordings.

2) Weekly virtual meetings: The DEPICT collaborating teachers, the CS fellows, and the implementation team had weekly virtual meetings during the academic year. The purpose of these meetings was to plan the in-class activities that were going to be implemented the following week. The high school teachers provided feedback on how each activity could be connected to their CW content. CS fellows provided feedback on the CT concept that each activity would target. The implementation team provided feedback related to the implementation task. All members decided on necessary modifications to generate a documented lesson plan.

a) Virtual repository: After every set of activities DEPICT researchers kept a virtual repository where they expressed in detail their impressions of the activity. Documents included a description of something that resonated with them about the lesson from the implementation; things that did not work or did not go as expected when infusing CT as part of their lesson; the benefits they saw in infusing CT within the lesson; any drawbacks they saw about the integration; suggestions to improve the implementation, and pictures from the activity.

E. Building the Curriculum

DEPICT4OST took as basics four main CT concepts (Decomposition, Pattern Recognition, Abstraction, and Algorithmic Thinking). For each student to have the experience on all the CT concepts, they were asked to develop a lesson for each of the four concepts, but they got to choose which CW or CM content to combine it with. The first lesson was going to be the umbrella for the rest, therefore, CS students took the most amount of time building the first lesson. The first draft was presented to the CS Outreach staff who gave them feedback on structure, ideas for extension activities, and time for each activity. A second draft we presented to the personal of the STEM outreach center who runs the After-School (A-S) programs in the local district. The final version of the curriculum was presented during the Spring Semester Kick off for A-S program.

IV. Undergraduate Experience in Apprenticeship

DEPICT researcher interviewed the CS undergrads about their impressions of the program. By participating in DEPICT, CS fellows mentioned different aspects that the program has impacted them.

A. Their Role as Facilitators of Learning

CS fellows start thinking as thought provoking facilitators that can transfer their knowledge into an inquiry based learning. "*My experience in DEPICT has changed the way I think about things… a lot of the concepts are really simple, but you can apply them… so I guess it's just going to change my perspective on things.*"

B. The Impact of Active Learning Techniques on Student Engagement

CS students described their impression about how high school students, particularly women, were making connections to computing, and were engaging differently, when completing activities. *"There's a lot of females in the classrooms that I've gone to and they are very engaged in the activities that we do, like the co-spaces. I've seen that they do the work that they're supposed to, and they try to improve their work. In CW, I've seen more women, and then all of them that go work on their projects… they do participate in the work."*

C. The Evolution of Their Teaching Practices

CS fellows learned that computational thinking can be applied to different real-world scenarios. The DEPICT implementation helped them see how the problem-solving skills that they regularly used in their college assignments were helpful to other disciplines. *"It helps me to think about different ways how to solve any problem, either in computer science or outside computer science, and that's just through talking with others, and looking at things differently in the classes. Or the way we teach the students, I also learn from it."*

V. Challenges and Lessons Learned

1) Adaptation of complex computational concepts for high school students: CS students were require to pull innovative teaching strategies such as using interactive demonstrations, real-world examples, project-based learning, and peer collaborations. Furthermore, CS student faced the challenge to explain the relevance and applicability of CT concepts in various contexts.

2) The alignment of activities with educational standards: when providing the training for (A-S) teachers, CS fellows focused on the CT concepts they were experts on. However, (A-S) teachers were always relating how the activities will merge with the educational standards.

3) Pedagogical insights gained: CS fellows discover an appreciation for the difficulties inherent in teaching that they had not initially anticipated. The CS students realized that learning involves more than finding the correct answer to a given problem. For instance, (A-S) students should be able to apply more than a specific kind of algorithm; they should analyse how to apply any type of algorithmic thinking process to provide an optimal solution.

4) Recognizing limitations and increased self-confidence: CS fellows should have mastered the material in order to provide meaningful guidance when developing the lessons. In addition to technical knowledge, CS fellows should possess strong communication and interpersonal skills to ensure that the future (A-S) teachers will know how to implement the lesson. CS undergrads were pulled from their introversion to being able to explain their written ideas to an audience that was not part of their technological domain.

VI. Conclusion and Future Work

The main goal of DEPICT4OST was to make abstract computational principles accessible and engaging through interactive writing exercises and hand-on activities. Each of the activities was designed with the purposed of leveraging the potential of CT to engage learners with minimal prior exposure to computing, fostering collaborative problem-solving skills. DEPICT4OST is being piloted in two (A-S) classrooms. As a next step CS undergrads will collect data to analyze the effectiveness of their curriculum. This step will pave the pathway for the CS undergrads who want to continue doing research in CS Education.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant #2137581. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] P. J. Denning and M. Tedre, *Computational thinking*. Mit Press, 2019.

[2] A. Bundy, "Computational thinking is pervasive," *Journal of Scientific and Practical Computing*, vol. 1, no. 2, pp. 67–69, 2007.

[3] S. Grover, "Computational thinking, programming and the google app inventor," *Kids and Technology*, 2010.

[4] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner, "Computational thinking for youth in practice," *Acm Inroads*, vol. 2, no. 1, pp. 32–37, 2011.

[5] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[6] C. L. Isbell, L. A. Stein, R. Cutler, J. Forbes, L. Fraser, J. Impagliazzo, V. Proulx, S. Russ, R. Thomas, and Y. Xu, "(re) defining computing curricula by (re) defining computing," *Acm Sigcse Bulletin*, vol. 41, no. 4, pp. 195–207, 2010.

[7] L. Perković, A. Settle, S. Hwang, and J. Jones, "A framework for computational thinking across the curriculum," in *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, 2010, pp. 123–127.

[8] L. Pollock, C. Mouza, K. R. Guidry, and K. Pusecker, "Infusing computational thinking across disciplines: Reflections & lessons learned," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 2019, pp. 435–441.

[9] V. Barr and C. Stephenson, "Bringing computational thinking to k-12: What is involved and what is the role of the computer science education community?" *Acm Inroads*, vol. 2, no. 1, pp. 48–54, 2011.

[10] G. H. Fletcher and J. J. Lu, "Education human computing skills: rethinking the k-12 experience," *Communications of the ACM*, vol. 52, no. 2, pp. 23–25, 2009.

[11] Authors, "Depict: Discovering computational thinking in creative writing courses," 2023.

[12] Code.org, CSTA, and E. Alliance, "State of computer science education: Illuminating disparities," 2020. [Online]. Available: https://advocacy.code.org/stateofcs

[13] B. Powell, "Five things the us can learn from china," *Time*, vol. 23, pp. 21–27, 2009.

[14] A. B. Tucker, "K-12 computer science: Aspirations, realities, and challenges," in *Teaching Fundamentals Concepts of Informatics: 4th International Conference on Informatics in Secondary Schools-Evolution and Perspectives, ISSEP 2010, Zurich, Switzerland, January 13-15, 2010. Proceedings 4*. Springer, 2010, pp. 22–34.

[15] J. Wang, H. Hong, J. Ravitz, and S. Hejazi Moghadam, "Landscape of k-12 computer science education in the us: Perceptions, access, and barriers," in *Proceedings of the 47th ACM technical symposium on computing science education*, 2016, pp. 645–650.

[16] J. Gal-Ezer and C. Stephenson, "A tale of two countries: Successes and challenges in k-12 computer science education in israel and the united states," *ACM Transactions on Computing Education (TOCE)*, vol. 14, no. 2, pp. 1–18, 2014.

[17] S. Hug, H. Thiry, and P. Tedford, "Learning to love computer science: Peer leaders gain teaching skill, communicative ability and content knowledge in the cs classroom," in *Proceedings of the 42nd ACM technical symposium on Computer science education*, 2011, pp. 201–206.

[18] E. C. Johnson, B. A. Robbins, and M. Loui, "What do students experience as peer leaders of learning teams?" *What Do Students Experience as Peer Leaders of Learning Teams?*, 2015.

[19] N. Santiago, M. Jimenez, and L. Guillemard, "Combining strategies for leadership development of engineering students," in *2020 ASEE Virtual Annual Conference*, 2020.

[20] J. Cuny, L. Snyder, and J. Wing, "Demystifying computational thinking for non-computer scientists (work in progress)," 2010.

[21] N. R. Council *et al.*, *Report of a workshop on the pedagogical aspects of computational thinking*. National Academies Press, 2011.

[22] P. J. Denning, "Computing is a natural science," *Communications of the ACM*, vol. 50, no. 7, pp. 13–18, 2007.

[23] D. Jonassen, "Using cognitive tools to represent problems," *Journal of research on Technology in Education*, vol. 35, no. 3, pp. 362–381, 2003.

[24] Y. Kafai, J. Griffin, Q. Burke, M. Slattery, D. Fields, R. Powell, M. Grab, S. Davidson, and J. Sun, "A cascading mentoring pedagogy in a cs service learning course to broaden participation and perceptions," in *Proceeding of the 44th ACM technical symposium on Computer science education*, 2013, pp. 101–106.

[25] K. Servais, M. L. Derrington, and K. Sanders, "Professional learning communities: Concepts in action in a principal preparation program, an elementary school team, a leadership team, and a business partnership." *International journal of educational leadership preparation*, vol. 4, no. 2, p. n2, 2009.

[26] C. Board, "Ap computer science principles curriculum framework," 2014. [Online]. Available: https://apcentral.collegeboard.org/courses/ap-computer-science-principles