CAREER: DRK-12: Spreading Computational Literacy Equitably via Integration of Computing in Preservice Teacher Preparation

Motivation

Computers are as transformational as the printing press. The printing press made information accessible to people who could read. As readers became more informed, many started to write and produce readings of their own, democratizing who were keepers and sources of information (McLuhan, 1962). For this reason, literacy is a fundamental skill that our society teaches every citizen. Computational literacy is quickly becoming just as fundamental (Wing, 2006). Just as readers became writers, the transformative power of computing is being able to create with a computer, by producing software, apps, or hardware that uniquely serves one's own needs. To advance equitably as a society, we must enable all members to gain computational literacy--to be able to create with computers.

In response to the demand for computational literacy, many public education systems are adapting as quickly as they can. However, they face a major bottleneck--a gap of computationally literacy in teachers. Most teachers, even the most junior preservice teachers, did not have access to computing education in their own K-12 education (Qian et al., 2018). To address this gap, many resources have been devoted to developing computer science (CS) or computational thinking (CT) activities that in-service teachers can be trained to use in their classroom with minimal computing knowledge.

These resources have increased access to computing education, but not equitably. Teachers who participate in these programs are primarily from districts that can afford to send them to professional development and do not need to adapt the activity to the context of their classroom (Margolis et al., 2015). Moreover, many resources are designed as elective, standalone CS activities or courses, meaning they are used mostly by students who seek them (Goode et al., 2014) and do not demonstrate how computing can be applied to almost any domain. To democratize computing education and show computing's vast applicability to most domains, *we need to incorporate computing into required courses* through activities that teach both the required subject area and computational literacy. Integrating computing across subjects can equitably spread computational literacy and harness the power of computing for teaching, learning, and achieving across domains.

To comprehensively incorporate computing into K-12 education, computing must become a part of all preservice teacher (PST) preparation programs. PST programs, however, are already some of the most content-packed and course-credit-laden undergraduate programs because PSTs must take subject area *and* education courses. Adding computing content (CT and CS) and pedagogy could add stress to already strained programs—the reason most computing educators have focused on in-service instead of PSTs for computing integration. This is no longer a viable justification, given the unequal spreading of computational literacy. Instead, we must study how to sustainably include computing in PST programs.

Research and Teaching Goals

Sustainable integration of computing into PST programs requires research on two fronts: 1) how to teach computing content and integration to PSTs in a way that builds upon their existing knowledge of their primary subject area(s) and teaching practices, and 2) which CT and CS concepts are most useful for completing activities in other subject areas? The former is a pedagogical, applied question about the best fit for computing in PSTs programs. The latter is a fundamental question about the nature of computational literacy. Because computational literacy is a new set of skills, there is still much debate about which CT and CS concepts are part of a basic or advanced literacy and to what degree concepts must be understood for citizens to apply them to improve their lives. Therefore, this project supports the spread of computational literacy by using iterative design-based research to

1. Incorporating computing into all PST programs at my university to explore sustainable models for computing integration (research goal) and to prepare future teachers to use computing integrated activities in their future practice to spread computational literacy (teaching goal).

2. Exploring the usefulness of various CT and CS concepts within the context of various PST programs, and, thus, exploring the usefulness of those concepts applied to a range of subject areas and grade bands to better define foundational computational literacy (research goal).

PI's Unique Position and Qualifications

I am one of a small percentage of computing education faculty who are in a college of education. To leverage this position, over the past three years I have been building support throughout my college to teach CT and CS in our PST preparation programs. A year ago, we piloted a college-wide initiative that will be fully implemented this year to include computing in all PST preparation programs. By 2021, all ~300 preservice teachers who graduate from Georgia State University annually, regardless of grade level or primary subject area, will be prepared to teach at least one extended, computing integrated lesson. In addition, the teachers will have practiced this lesson through practicum or student teaching, increasing the likelihood that they will be comfortable implementing it in their class. This widespread computing integration creates an ideal environment in which to study computational literacy.

My research context is also ideal because I am the director of the Computer Science Endorsement program at Georgia State, which prepares pre- or in-service teachers to add computer science to their teaching certification in another subject. The endorsement program provides a powerful comparison group. The endorsement teachers will be developing a deep understanding of computing concepts in order to teach standalone CS courses, but they also teach other subjects. Thus, they will be engaging in computing integration throughout their program as well. Working with both groups of teachers uniquely positions me to explore the role of teachers' depth of computing content knowledge on computational literacy and application of computing integration activities. Based on these roles in my university, the research and teaching goals of the proposed project are inextricably intertwined.

In addition to my roles in my university, my position is optimal because I work closely with the Georgia Department of Education (DOE) on their CS initiatives. My contributions include being part of the team who wrote the K-8 CS Standards that are now adopted in Georgia and the team making recommendations for implementing the standards, much of which is through computing integration. Therefore, I am well-versed the state's requirements for CS, which informs my work with PSTs. Furthermore, being on these teams and part of their CS Advisory Council keeps me in regular contact with practitioners, administrators, and policy makers to align goals in K-12 computing education.

Intellectual Merit: To address my two research goals, I will answer three research questions:

RQ1: What similarities do PSTs recognize between their primary subject area(s) content knowledge and teaching practice and computing content knowledge and teaching practice (i.e., computing integration)?

RQ2: What computing knowledge do PSTs have after the intervention (i.e., completing an hour-long CT module, engaging with a computing integration activity and CS content, and applying the activity)?

RQ3: How does computing knowledge affect confidence and quality of computing integration activities, especially in terms of subject area and computing learning objectives?

Though other researchers are currently addressing these questions, my unique context allows me to answer them with a breadth (i.e., across grade bands and subject areas) and depth (i.e., iteratively and compared to teachers who are preparing to teach computer science) that will contribute significantly.

Summary of Broader Impact: I devote my research and teaching to computing integration because integrated computing is an equitable approach to increasing computational literacy. I prioritize equity as does my university as a minority-serving institution and my city as a majority-minority city. By conducting this work at Georgia State University with PSTs who will most likely work in the Atlanta area, I am ensuring that my efforts benefit those who are underrepresented in computing, specifically teachers and students who are from low-income families and predominately African American. My efforts also align with my college's emphasis on culturally relevant pedagogy and social justice.

Background

CT emerged as a framework for integrating computing concepts and practices into other subject areas. The primary benefits of CT depend on who you ask, ranging from **K-12 educators** emphasizing computers as a tool for learning (Kale et al., 2018), **K-12 education researchers** emphasizing the affordances of computing technology (DiSessa, 2014; Tofel-Grehl et al., 2017), **computing education researchers** emphasizing the equitability of engaging more students in computing (Grover & Pea, 2013), and **computer scientists** emphasizing the power of computing for solving many kinds of problems (Aho, 2012). All perspectives are valid and used to motivate integration of CT in K-12.

Perhaps because so many stakeholders have become invested in CT integration, which skills constitute CT have been continuously debated since Jeanette Wing first popularized the term. Wing's (2006) definition, which has largely fallen out of favor, described CT somewhat vaguely as solving problems like a computer scientist would. Some have built upon this definition by describing *how problem solving in CS aligns with problem solving in other subjects* (Barr & Stephenson, 2011; Weintrop et al., 2016). Others have built upon this definition by describing the *higher order thought processes* that are trained in CT (Aho, 2012; Denning, 2017). In my opinion, the most useful expansions on Wing's definition, based on my context, are those that have defined CT as a *literacy that comprises both CS concepts and practices and enables students to use computers as tools, no matter the academic or non-academic domain* (Armoni, 2015; Brennan & Resnick, 2012; Grover & Pea, 2013). Within the group that shares this view, there is still contention about which concepts and practices should be included. For example, in the first half of 2019 alone, *Communications of the ACM*, which is not even an education publication, three blogs have discussed this issue (Guzdial, 2019; Schanzer et al., 2019; Tissenbaum et al., 2019). This debate stems from not knowing which concepts and skills are necessary for basic computational literacy.

This project builds upon existing work to better define basic computational literacy and to explore sustainable methods for incorporating this literacy into PST preparation programs. As a computing education researcher who works in a college of education, I am well suited to build upon prior work from both computing education and K-12 education because I regularly communicate with colleagues from both communities. Therefore, I know how prior work has been received in the communities and what are good areas for improvement. On the computing education side, for example, I know that the K12CS framework is viewed as not rigorous enough by many computing education researchers for standalone CS. I can use this context as well as that for the Computer Science Teachers Association standards for K-12 computing and the Learning Trajectories from Everyday Computing, which breakdown computing concepts into subcomponents and map out which components are prerequisite for more advanced components (Rich et al., 2018), to define concepts that the computing education research community values. On the K-12 education side, I know that CT and computing integrations can often be perceived as one more thing that is stuffed into the curriculum, but they are often viewed more favorably if the teacher preparation learning outcomes are emphasized. I can use this context to inform effective implementation of various resources for computing integration activities, including those from NSF STEM+C grants, Google's CT materials database, the Center for CT at Carnegie Mellon, simulation-based activities from PHET and NetLogo, event-based activities from Scratch and MIT Android App Inventor, and others.

The theoretical foundation of this work is based on constructivism. Constructivism states that learners gain new knowledge by building upon their prior knowledge. Thus, constructivists believe that educators should encourage learners to explore new concepts and skills and how they relate to concepts and skills that they already know rather than directly tell them what to learn and how to learn it (Tobias & Duffy, 2009). Constructivists do not support unguided learning, though. Instead, they see educators' role as providing scaffolding and guidance when needed (Hmelo-Silver et al., 2007). The interventions I have designed rely on constructivist techniques to prompt PSTs to recognize connections between the new CT and CS concepts to prior knowledge in their subject area(s). My role as an educator is to provide scaffolding to support learning the CT and CS concepts. Because PSTs have little prior experience with computing, the instruction is on the more guided side of constructivism.

Current Work

At Georgia State, I have been working with PST preparation faculty to incorporate computing into our programs. Our implementation has three parts. The first part is a general CT instruction module used in one of the first education courses all PSTs take. The module is domain-inclusive and provides examples from all domains. Applying constructivism, it asks PSTs to connect the concepts that are included (i.e., abstraction, algorithms, automation, deconstruction, and debugging) to the subject(s) in which they are specializing. To facilitate connections, PSTs are grouped based on grade band and subject so that they may discuss applications and similarities with their peers.

The second part is practicing a computing integrated activity in their methods courses with just-in-time CS content (e.g., variables and loops) required for the integration. I guest lecture during one week of class to introduce the activity and help the PSTs become comfortable with it. I select, create, or customize activities based on topic requests from the faculty instructor to ensure that the activity fits the needs of the class. Our goal for the PSTs is not that they create new activities, but that they are comfortable adapting the plethora of existing activities for use in their class. Thus, I scaffold the first activity by providing a pre-built, customized activity and show them resources for other activities. After practicing the activity, the faculty instructor and I help PSTs develop a lesson plan to use the activity in student teaching or practicum, which is the third part of the implementation.

From the pilot year, I have found that teachers are initially hesitant but later excited to use computing to power creativity and differentiation in their lesson plans and to provide opportunities for their students to learn about computing. However, teachers are also anxious about their depth of CT and CS knowledge, particularly stemming from uncertainty about helping their students to debug and from general expectations that CS is highly difficult or not something they cannot understand. By exploring the bounds of computational literacy, I hope to give them evidence-based expectations of what constitutes rigor and competency in computing.

Sample Computing Integrated Activity:
During the pilot year, we used digital modeling as a computing integration in middle science. A digital model built in Pencil Code (see Figure 1) provided a dynamic visualization of a wave in which students can change the values of different wave properties, like the wavelength, to explore how it affects the wave. The model also provides opportunities to teach computing. Students can learn about computing concepts such as defining and using *variables*, defining and calling *functions*, and using *loops* by exploring the different pieces of code and how they contribute to the model.

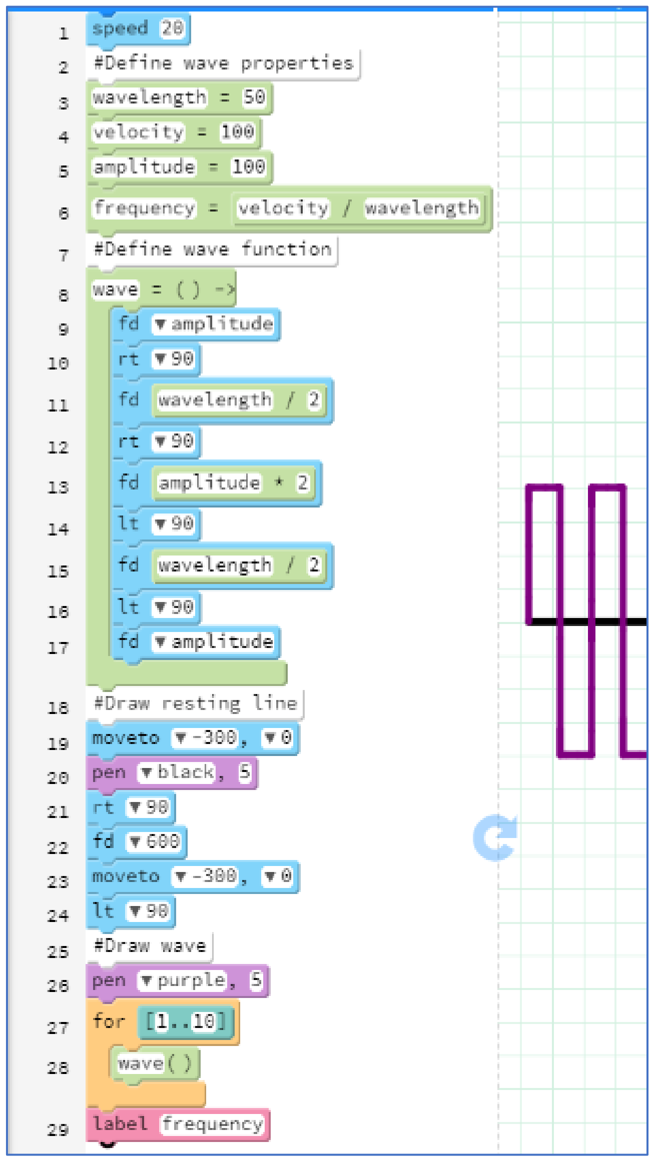

Figure 1. Dynamic model of a wave created with Pencil Code for middle science methods.

Proposed Research: Design and Preservice Teacher Intervention

The proposed research project uses the context of Georgia State's PST programs to address the research goals: to explore sustainable models of incorporating computing into PST program and to examine which CT and CS concepts are most foundational to a general computational literacy. To achieve these goals, I will use design-based research because it is an iterative methodology that begins in a specific context and gradually generalizes to a range of contexts. This approach is appropriate for an area of research that still has many unknowns and that I plan to conduct within Georgia State's programs before generalizing to other teacher preparation programs nationally after the proposed project. For example, during the pilot year, I included variables, functions, and loops as CS concepts in the model in Figure 1. The PSTs, however, struggled to understand the function in the model. Two likely explanations are that 1) the model did not need a function to work efficiently, so the PSTs had trouble understanding its purpose, or 2) functions are a difficult concept for all CS students, and they might not be necessary for a basic computational literacy. By working iteratively within the PST programs and using the CS Endorsement teachers as a comparison group, I can explore these possibilities.

Design-based research also fits well with my plan to mix qualitative and quantitative data collection and analysis. Earlier in the exploratory stage of the project, more emphasis on qualitative data allows the researcher to better understand the dimensions of intervention's effect. As iterations progress and the researcher's understanding solidifies, more emphasis on quantitative data allows the researcher to examine the size of effect on each dimension. For example, during the pilot year, I asked PSTs to reflect on their experience during the CT module, computing integration activity, and student teaching. Many of them mentioned being uncertain of their computing abilities. Based on that qualitative data, I can include a quantitative self-efficacy instrument to examine their self-efficacy through the intervention.

In the current iteration, the general CT module includes an introduction to CT and computing integration and five skills (see Figure 2). These skills were selected based on discussions with Georgia State faculty and definitions in the literature (Aho, 2012; Armoni, 2015; Barr & Stephenson, 2011; Brennan & Resnick, 2012; Cuny et al., 2011; Denning, 2017; Grover & Pea, 2013; Weintrop et al., 2016; Wing, 2006, 2008). Each component includes definitions, examples from various academic and non-academic domains, and related computing concepts, such as conditionals in algorithms and serial or parallel processing in automation. The components include reflection questions that prompt PSTs to relate the topics to their subject area, which they discuss within groups based on grade band and subject. These discussions will ensure that the PSTs voices are represented while exploring computing integration in PST programs and computational literacy concepts.

Figure 2. List of CT concepts introduced in the domain-independent module.

1. What is CT and motivation for computing integration; CT defined as answering three questions:
   i) Should I get a computer to help me solve this problem (is it a good tool in this case)?
   ii) How would I get a computer to solve this problem?
   iii) Does the computer solve the problem accurately and efficiently?
2. Abstraction
3. Algorithms
4. Automation
5. Deconstruction
6. Debugging

During the second step of the intervention, PSTs apply these general CT skills to a grade- and subject-appropriate computing integration in their methods courses (see Figure 1 for middle science). Because most PSTs have no CS background, CS content knowledge is provided just-in-time based on the 4C/ID model (van Merrienboer, 2004). The 4C/ID model is an instructional design method for teaching complex skills, like computing integration, with authentic problems. As prescribed by the 4C/ID model, the PSTs first engage with a scaffolded or simplified version of the problem. For the model in Figure 1, PSTs started by learning a simplified model that included only wavelength and not amplitude. After mastering that version, they then added amplitude to the model. Based on the pilot year, this year we will start with a simpler version without variables (i.e., fd(50) rather than fd(wavelength)). Then we can introduce the value of variables meaningfully instead of telling the PSTs that it is more efficient. We will also exclude the concept of functions. To design activities, I work with the faculty who regularly teach the methods courses to select suitable topics for computing integration, ensuring that the activity aligns with standards that apply to that subject, like NGSS for science, and fits within the objectives for the course. Toward the project, the computing integration in a range of methods courses provides opportunities to explore both depth of CT and CS concepts to authentic problems in other subjects and breadth of applicable concepts across a range of subjects and grade bands to address the research goals.

The last step, implementing a computing integrated lesson plan in student teaching or practicum, keeps the project tethered to the demands of authentic teaching practice. The project would not serve to spread computational literacy if the activities that we practice in the programs are inaccessible to students or if the PSTs never take the final step of using them in their classroom. Though the scope of the project does not include learning outcomes for students, this step provides additional benefit by introducing current students to computing integrated activities. In the pilot year, most of the students who we worked with during student teaching were excited to learn about computing and to keep working with Pencil Code on their own. Their excitement had a reinforcing effect on the PSTs, making the PSTs excited to continue to teach computing integrated activities. Many PSTs, after student teaching, had ideas for other computing integrated activities that they wanted to implement in math (e.g., creating graphs), history (e.g., manipulating data), or English (e.g., analyzing poems).

Proposed Research: Timeline

Activity	pre-proposal	2020-21 Year 1	2021-22 Year 2	2022-23 Year 3	2023-24 Year 4	2024-25 Year 5
Iterative design of CT module	X	X	X			
Iterative design of computing integration activities	X	X	X	X	X	
Primarily qualitative data analysis	X	X	X			
Primarily quantitative data analysis				X	X	X
Dissemination	X	X	X	X	X	X
Visit other computing preparation programs		X	X			
Prepare for scale-up					X	X

Proposed Research: Participants and Comparison Group

The primary participants in the proposed research project will be PSTs in the initial teacher preparation programs at Georgia State. Each year, we have about 300 new teachers enter programs in elementary, middle, and secondary education. Our PSTs represent the student population at Georgia State, which includes 42% African American, 25% White, 13% Asian, 10% Latinx, and 10% of other or mixed races. Our gender distribution is balanced with 56% female students. Nearly 40% of our students are first-generation college students, and almost 60% of our students are on Pell grants (i.e., from low-income families). PSTs in elementary programs will take the CT module in the early childhood and elementary classroom management course, which is taken during the first semester in the major. Then they will engage with computing integration activities in their math methods and science methods courses at both the K-2 and 3-5 grade level (i.e., four courses and activities in total). PSTs in middle and secondary programs will take the CT module in the middle and secondary opening school experience course, which is taken during the first semester. Then they will engage with computing integration activities in either their math, science, social studies, or English language arts methods courses at either the middle or secondary level. Middle grades PSTs specialize in two subject areas, so they will likely engage with two computing integrations. All methods courses have an associated student teaching experience in which PSTs will use the computing integration activities unless the school requests something else.

During student teaching, I will collect only observational data about the students engaging with the computing integration. Students are not intentionally participants in the proposed research, but their interactions with the PSTs and activity will inform the iterative design of the integration activities.

The other group of participants, teachers in the CS Endorsement program, serve as a comparison group. The endorsement program adds on the area of CS to a teaching certificate in another area, enabling teachers to offer standalone CS courses. This year, '19-'20, is the first year of the program, but we expect to have about 10 candidates. We also expect the number to grow each year based on a new law that requires all middle and high schools to have standalone CS courses by 2025 and provides state-funding for teachers to earn a CS credential. Despite this new requirement, schools typically do not have enough demand for CS to hire a teacher who teaches only CS. Thus, we designed the endorsement program for in-service or preservice teachers to add CS to another area, and one of the four courses that they take will focus on CT and computing integration, and they will teach a computing integration activity in their class. Therefore, this group, as they gain a deep understanding of computing concepts, provides a small but powerful comparison group to address the research questions.

Proposed Research: Research Questions

Throughout the five years of the proposed project, I will iteratively explore three questions.

RQ1: What similarities do PSTs recognize between their primary subject area(s) content knowledge and teaching practice and computing content knowledge and teaching practice (i.e., computing integration)?

The answer to this question will allow educators to prepare and use computing integration activities based on natural connections between their subject areas and computing, as identified by teachers. I decided to start the intervention with the general CT module that emphasizes connections to PSTs' primary subject area(s) because I want the PSTs to build upon their content and pedagogical knowledge in their primary subject area(s). This approach also ensures PSTs give input into how they would like to use computing in their subject, increasing the sustainability of computing integration and providing non-computer-scientists' perspectives on important computational literacy skills. For example, based on the pilot year, science PSTs wanted to be able to automate data collection and analysis, so this year we will be using a data collection and analysis computing integration in 3-5 science. The limitation of this approach is that PSTs are identifying computing integration opportunities before they have learned much computing, which makes the iterative nature of the proposed research important. The iterations allow for multiple cycles of PST input balanced with design of activities by a computing educator.

RQ2: What computing knowledge do PSTs have after the intervention (i.e., completing an hour-long CT module, engaging with a computing integration activity and CS content, and applying the activity)?

The answer to this question determines how much time needs to be devoted to computing education in PST programs to produce computationally literate teachers who will equitably spread computational literacy. The requirements for PST programs are already demanding, and I am fortunate to have support to incorporate computing education into Georgia State's programs. I am maximizing the time I have with teachers by building on their prior knowledge, spreading the intervention over several semesters, and ensuring that they apply their knowledge authentically. We need to know whether this is sufficient. We can also compare PSTs' knowledge to that of teachers in the CS Endorsement program. Based on the pilot year, the PSTs changed their view of technology from consumers of technology (e.g., using websites or

Word) to producers of technology (e.g., write programs and automate tasks) over the intervention. They provided robust definitions of the CT skills, had confidence that they could apply pre-built computing integration activities in their class, **and described how they would like to apply computing integration to teaching other subjects, though they were never prompted to make these connections**. However, they were still uncertain of their ability to write and debug programs. During future iterations of the intervention, I will attempt to increase their programming and debugging skills. However, given the plethora of computing integration activities and resources for debugging, these skills might not be as critical as knowing where to find help, as addressed in RQ3.

RQ3: How does computing knowledge affect confidence and quality of computing integration activities, especially in terms of subject area and computing learning objectives?

The answer to this question informs the level of computing education that non-computing teachers need to effectively integrate computing into their primary area and spread computational literacy. Based on my context in which I work with both PST programs and the CS Endorsement program, I can explore the differences between computing integration based on depth of computing knowledge. This question also addresses which computing concepts are most important for a basic computational literacy. If the CS Endorsement teachers have learned control structures but use only conditionals and loops in their computing integration activities, then perhaps functions are only important for more advanced computing concepts. Not that functions are not applicable to problems in other subject areas, but they might not increase efficiency enough to warrant including them in all PST programs. The range of PST programs in the study and the comparison group of CS Endorsement teachers allows me to uniquely address this question and the others to contribute to knowledge on this topic.

Proposed Research: Data Collection and Analysis

Data Source	RQ(s)	Collection Method	Collection Time
CT survey	1, 2, 3	Qual & quant	Pre-CT module, post-CT module, post-methods class, post-practicum
CT module reflections	3	Qualitative	During CT module
CT module quizzes	2	Quantitative	During CT module
Post-class assignment	1, 2, 3	Qualitative	Post-methods class
Lesson plan	3	Qual & quant	Post-methods class
Post-practicum reflection	1, 3	Qualitative	Post-practicum
Post-practicum quiz	2, 3	Quantitative	Post-practicum
Field observations	1, 3	Qualitative	During methods class and practicum

Table 1. Data sources, research question(s) addressed, data collection method, and collection time.

Most of these data sources were used in the pilot, but the CT module quizzes and post-practicum quiz were added this year to better measure computing content and pedagogical knowledge. The **CT survey** was based on the survey from Yadav et al. (2014). The survey has a quantitative component that asked PSTs to rate 1) their familiarity with CT (pre-test only), 2) how easily CT can be integrated into other subjects, 3) how comfortable they would be integrating CT, and 4) their general comfort with using computers. Based on the pilot, a short self-efficacy instrument will be added. The qualitative component of the survey asks PSTs to explain/define 1) CT, 2) how they might implement CT in their class, 3) barriers that they might face implementing CT, and 4) list three things that someone who knows computing could do. The **CT module reflections** are PSTs responses to the reflection questions at the end of each CT skill, as described above. The **CT module quizzes** are three-question, multiple-choice quizzes added at the end of each of the five CT skills. They were designed to include distractor questions that might capture misconceptions.

The **post-class assignment** is completed after engaging with computing integration activities in methods classes. The assignment has several components: 1) reflection on classroom instruction, 2) explain CT and the five skills discussed in the CT module, 3) reading reflections on articles about CT written by Kale et al. (2018) and Grover and Pea (2013), 4) expand/modify the program introduced in class to add/change a feature of the computing integration activity, and 5) match the computing integration activity to subject area, ISTE, and social justice standards. The **lesson plan** is also completed after the methods class. The PSTs use a lesson plan template to write a 90-minute lesson plan for using the computing integration activity in preparation for practicum or student teaching.

The **post-practicum reflection** is due two weeks after student teaching or practicum. The reflection asked four questions: 1) What parts of the activity went well, and what parts did not go well?, 2) What would you change about the lesson plan based on your experience?, 3) What are the trade-offs in using a computing integrated lesson?, and 4) If you started with a pre-built resource, like the activity introduced in class, how likely and comfortable would you be using a computing integrated activity in the future? The **post-practicum quiz** will be new this year, and it will be customized based on the computing concepts used in the activity. When possible, I will use questions from validated tests by searching CSEducation.org. Using questions from validated tests does not mean that the quiz will be validated, but the question will have been through some vetting.

To supplement the data collection from PSTs, I take field notes during the classroom instruction and practicum or student teaching. Because I am engaged in instruction or facilitation, these are not systematic field notes from an impartial observer. Instead they are to record the topics discussed in the classroom instruction and anything of note, either good or bad.

Each research question has at least four sources of data collected at various points throughout the intervention and a mix of qualitative and quantitative data collection methods. In the early years of the project, the qualitative data will be analyzed using qualitative methods to explore the dimensions that are most salient and impactful. For example, to analyze data from the pilot, I used content analysis (Hsieh & Shannon, 2005) supported by NVivo 12. Content analysis allows themes to emerge from the data by iteratively coding the data to explore different interpretations. The initial, exploratory nodes that I started with were to code which research questions the datum addressed. During the first round of analysis, I classified PST responses into the research question nodes and made additional nodes for high-level themes within each research question. During the second round, I classified responses within each research question into the high-level themes and made additional nodes for sub-themes. During the third round, I classified responses within the themes into sub-themes and did not recognize additional thematic nodes, suggesting saturation based on the research questions.

As the project progresses, the qualitative data will be analyzed with more quantitative methods to examine the effect size of the intervention. For example, in the pilot I was interested in how the PSTs described applying computing integration to other subjects they might teach, but by year three of the project, I will likely be more interested in the proportion of PSTs who give other subject applications and whether the applications are the same as those suggested in previous years. Similarly, the lesson plans will be qualitatively analyzed in the earlier years and based on a rubric in the later years.

I will also collect data with quantitative measures. Based on the sample size, about 300 PSTs per year, I will be able detect an effect size as low as 0.1 by using repeated-measures, inferential statistics, based on G*Power ($\alpha = .05$ and power = 0.8). I will use parametric tests that are appropriate for the analysis (e.g., MANOVA for multiple dependent variables that represent the same construct vs. ANCOVA for co-variates) if the data meet the assumptions (i.e., continuous data, equal variance among groups, and normal distribution based on histogram, skewness, and kurtosis). If the assumptions are violated, then I will use appropriate non-parametric tests (e.g., Mann-Whitney instead of t-test for non-normal distributions). I will base my conclusions primarily on effect size rather than significance testing for more precision, but I will report both in dissemination. I will not be able to compare the PSTs to the CS Endorsement teachers using inferential statistics based on the low sample size of the latter group, but I can compare them using qualitative data and with descriptive statistics.

Proposed Research: Feasibility and Limitations

Because the proposed research is within an ongoing project, it has high feasibility. Georgia State is committed to incorporating computing into our teacher preparation programs, not just for PSTs. The computing intervention and associated assignments will be part of normal teacher preparation whether the proposed research is conducted or not. Our Associate Dean who is responsible for teacher preparation programs internally funded the pilot year of this project to incentivize faculty to engage in it. Furthermore, my college just submitted a US DOE Teacher Quality grant for incorporating computing into our teacher residency program, and the project was not even my idea. That proposal was led by our Associate Dean who is responsible for partnerships with local schools, who helps us find student teaching opportunities and advertises the skills of our teachers. In addition, I have met many leaders of computing-focused charter schools in Atlanta through my service work with the Georgia DOE who can provide opportunities for student teaching and who might hire our graduates. My work with the Georgia DOE also ensures that I am familiar with the latest policies and strategies for increasing computational literacy in the state, which I share with our teacher preparation faculty to help them prepare teachers who will be competitive. Based on the pilot year, the faculty engaged were more motivated to continue this project after the first application. I do not believe the enthusiasm for this project will fade.

This project reaches a wide range of PSTs, which has benefits that are unique in the current research but also limitations. While it allows me to explore computational literacy across topics, it also means that I do not have as much time to devote to any single topic. Moreover, analyzing qualitative data from 300 PST each year will force me to focus on the areas of alignment between teachers rather than the areas of divergence. Therefore, this project will tend to focus on what the participants have in common rather than potential areas for differentiation. I hope that I can engage other faculty and graduate students to work within this project, but I recognize that they have their own research interests. The proposed grant would allow me to hire a long-term graduate student specifically focused on this topic to help analyze the copious amounts of data I will be collecting.

Proposed Research: Expected Outcomes and Possible Alternatives

Related to the research goal to sustainably integrate computing education into PST preparation programs: I expect that the computing intervention, when spread throughout the preparation programs and practiced in student teaching, will prepare and motivate teachers to use computing integration activities in their class. However, they might have too little confidence or knowledge to make sustainable change that affect long-term implementation. In this case, more instruction or more scaffolding for the computing integration activities could be added, especially as e-kits for teachers to use after they graduate. Continuing PD for in-service teachers could also ensure that the activities are used in the classroom. The value of the project's iterative approach is that I have five years to explore options and examine their effect on quality and sustainability. After working within this context, I can take the best practices forward to scale beyond the programs at my university. Based on the breadth of the interventions, I might also be able to identify trade-offs and decision points for future implementations.

Related to the research goal to better define foundational computational literacy: Through this iterative work across many subject areas, I expect to find a common set of CT and computing skills that apply to most subjects and form a basic computational literacy. I expect that this set of skills will include some of the concepts used in existing work, like Grover et al.'s (2019) VELA (variables, expressions, loops, and abstractions) framework, but might also exclude other introductory computing concepts, like functions. I might also find that some concepts, like variables, are developmentally inappropriate for elementary school, but needed in later grades, and some concepts are easily taught in elementary school, like loops and conditionals. It is possible that one set of skill does not naturally apply to different subjects and instead teachers should focus on a subset of skills with their subject. Then the students would gain a complete computational literacy by engaging with different integrations in a variety of classes.

Evaluation and Mentorship

To guide this work and increase its success, I have requested the support of two experienced and influential computing education scholars who are involved in computer science teacher preparation and computing integration at their own institutions. I am calling each of them mentors to recognize that they serve as more than advisory board members by providing feedback on research, teaching, and professional development. The first is **Aman Yadav**, a Professor in the College of Education at Michigan State. Dr. Yadav is a preeminent, NSF-supported researcher of computing integration in K-12 who has recently started to focus on computing integration in preservice teacher programs. As a fellow psychologist and faculty in a college of education, he is an ideal mentor for me in this project.

My second ideal mentor shares many of my methodological and analysis interests. **Brian Dorn**, an Associate Professor in the College of Information Science & Technology at University of Nebraska Omaha, is one of the computing education researchers who also identifies as a learning scientist. He is knowledgeable in the theoretical and methodological backgrounds of my work and involved in several K-12 computing research projects, including NSF funded work, and teacher preparation programs. Please see Figure 3 for the logic model that will guide evaluation.

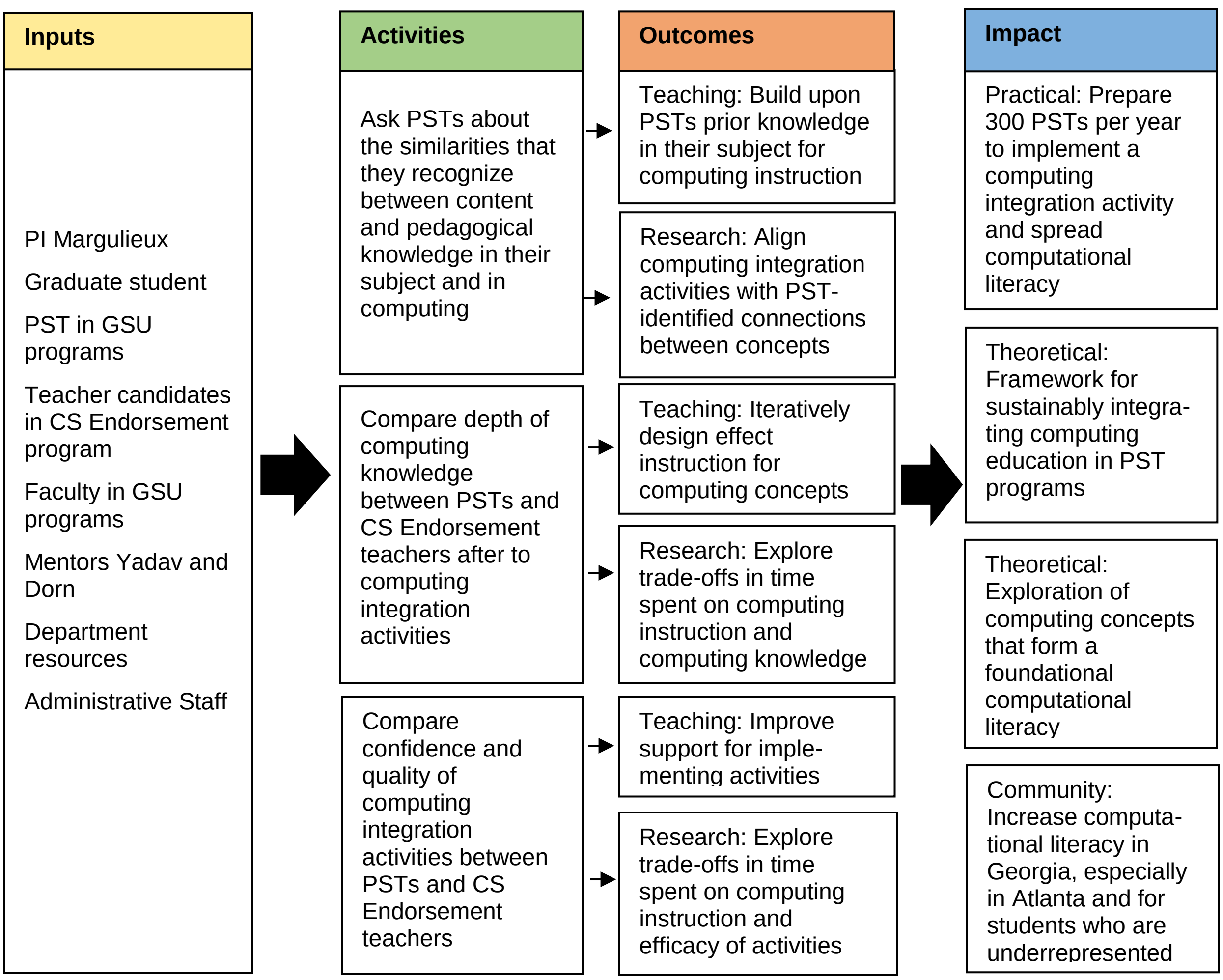

Figure 3. Logic model of activities and outputs for the proposed project.

I will visit the home institution of each mentor during the first year to learn about their computing teacher preparation programs, discuss the exploratory stages of my project, and plans for later stages. In the last four years of the project, Dr. Yadav will visit Georgia State during year 2 and 4, and Dr. Dorn will visit during year 3 and 5. During 2-day visits, I will update them on the progress of the research and teaching components, walk them through my methods, data, and analyses so far and request their feedback. In addition to these evaluation visits, I see both of them at least twice per year at conferences. We will schedule a time for all of us to meet annually at a shared conference.

Dissemination

Conferences and site visits are central to my dissemination and network building plan. I plan to submit to and attend at least two conferences per year. The conferences that I plan to attend regularly are described in Table 2. Because my intended audience for dissemination are people involved in teacher preparation programs, computing education researchers, and learning scientists, conferences are an appropriate and effective way to reach them. My publication plan is in Table 3.

Conference	Community	My Submissions
American Educational Research Association (AERA)	Teacher prep and education researchers, including groups focused on computing education and learning sciences	Practice-driven papers that focus on implementing activities in the classroom
International Conference of the Learning Sciences (ICLS)	Learning scientists in general education and discipline-specific education	Theory-driven papers that focus on computational literacy and applications in various subjects
International Computing Education Research (ICER) conference	Computing education researchers focused on improving theory	Theory-driven papers that focus on teaching computing in preservice teacher programs
ACM SIG Computer Science Education (SIGCSE)	Computing educators and researchers focused on improving practice, including for K-12 teacher preparation	Practice-driven papers that focus on computing integration in teacher preparation programs

Table 2. Conferences for Dissemination including Alignment of Community and Submissions.

In addition to in-person dissemination through conferences and site visits, I will publish my results in journals to share my research with the community following the publication strategy in Table 3. I have experience publishing as the first and corresponding author in high quality journals that would be relevant to this project, including *The Journal of the Learning Sciences*, *Learning and Instruction*, *Instructional Science*, *Computer Science Education*, and *Educational Research Review*. I also have experience publishing with graduate students, including recent experience in which a graduate student was first author. I will continue to seek publication in these journals, teacher preparation journals, such as *Teaching and Teacher Education*, and opinion and policy-oriented magazines, such as *Communications of the ACM*. Whenever the faculty who I am working with are interested in co-authoring papers, either for the publications planned in Table 3 or in their own subject-area publications, I would be thrilled to work on papers together to better reach an interdisciplinary audience. Likewise, if graduate students other than the GRA supported through the proposed grant want to use data collected through the project to address their own research question, I would be happy to help them plan and conduct suitable analyses and write papers based on the project.

I will further build my network and promote my work when I visit other universities to learn about their teacher preparation programs. In addition to visiting my mentors' institutions, I am requesting funding to visit two more universities that are integrating computing into their teacher preparation programs during the second year of the project, likely University of Maryland and Indiana University. For the visits, I will offer to give a talk about my work at Georgia State to interested faculty and students. These talks will allow me to disseminate my work and practice communicating about my work with different audiences.

Topic of Papers	Community	Year
Global design of computing integration, identifying needs for content to add and replace, and getting faculty buy-in	Teacher prep faculty and CS ed researchers	2020
Subject-area-specific integration activities and qualitative effect on computing knowledge and subject area teaching practice	Discipline-based education researchers	2021
Finalized global CT module based on 3 years of implementation	Teacher prep faculty	2022
Short-term efficacy and qualitative effect of intervention as PST cohorts graduate	Teacher prep faculty and CS ed researchers	2023
CT and CS concepts used widely and effectively in integrations	CS ed researchers	2024
Finalized integration activities based on 5 years of implementation	Teacher prep faculty	2024
Long-term sustainability of computing integration	Teacher prep faculty	2025
Quantitative effect of intervention on computing knowledge in teachers and quality of implementation	Discipline-based education researchers	2025

Table 3. Topical Publication Plan during Proposed Project.

Plan for Continuing Work after the Grant Period

During the proposed project, I will focus on computing integration within Georgia State University's PST and CS Endorsement programs. The range of programs in which I will be working provide substantial breadth for this stage of the project, and the five-year timeframe allows for iteration and depth of exploration and examination. While the population of Georgia State and Atlanta are particularly suited to spreading computational literacy to groups who are underrepresented in computing, the next stage of the project will need to scale-up beyond one institution. In the final years of the proposed project, as the iterative process settles on finalized designs, I will make plans for packaging the intervention for use in other programs in the US. These plans will be informed by my visits to other education programs during the first two years of the proposed research, discussions with faculty from other institutions at conferences, and feedback from my mentors. Each institution has a unique set of needs, especially relating to the motivation, skills sets, and values of the faculty who work there. Therefore, the intervention will need to be customizable based on the needs of each institution for effective scale-up.

One possible application of this work relates to another project I am involved in that is in the initial planning stages. There is a lot of interest nationally for developing microcredentials for teacher certification in computing education. Because computing teachers tend to be primarily trained in another subject area and because many short-term options for teacher PD about computing are available, a microcredential-based system allows teachers to demonstrate a cohesive knowledge of computing even if they have learned content in bits and pieces. At Georgia State, we have a Board-of-Regents-approved system for granting course credit for microcredentials, so I am interested in developing a set of microcredentials that would provide a credential for computing integration. While this idea is still developing, several organizations are enthusiastic about this approach. My college is considering funding the development of some of these microcredentials to create a proof-of-concept. The Georgia DOE is interested in developing short courses through their virtual PD to help in-service teachers fill in gaps in their knowledge to achieve the microcredentials. The International Society for Technology in Education (ISTE) is also interested in partnering with us to create national, microcredential-based, certification for computing integration. This project could inform the computing concepts that should be required for this microcredentialing system and the level of computing instruction that would be necessary to teach those computing concepts to non-CS teachers.

Research and Teaching Integration

The proposed project directly supports my teaching activities in several ways. The largest impact is on the PST preparation programs at Georgia State. Though my teaching responsibilities do not include teaching in these programs, I am honored that the faculty in these programs have entrusted me to share their limited time with PSTs to engage in computing education. I believe that working with teachers in these programs allows me to make the biggest impact on computational literacy that I can in my current position and at this point in my career. Thus, despite not being part of my teaching responsibilities, I am motivated to teach computing in these programs and design activities that work best for our teachers and their students. I hope to receive funding through this proposal so that I have more time and resources to conduct research within this context and improve upon my teaching efforts.

Outside of the PST preparation programs but still in teacher preparation, the proposed research also benefits the CS Endorsement program that I direct. I teach two courses in the program, a digital and information literacy course and a computational thinking course that uses block-based programming. These are the first two courses in the program, which build up to a text-based programming course that prepares candidates to teach AP CS A, which is an upper-high-school-level CS course. Because the CS Endorsement adds on an endorsement to certification in another subject area, both courses that I teach are designed to bridge candidates existing subject area knowledge to relevant computing concepts and to build skill in using foundational computing concepts. Therefore, they are full of opportunities for integration with other subjects. Many activities in the courses ask candidates to design computing applications in their primary subject, and the field experiences at this level can be completed with computing integration activities in their own class for candidates who are in-service. This project will improve the quality of my teaching in these courses as I explore the best ways to make connections between computing and other subjects. Furthermore, the candidates' applications of computing with their students will inform the research, especially the feasibility of computing integration activities and how students interact with them.

The proposed project supports teaching objectives for undergraduate and graduate students learning to conduct research. The most reliable benefit would be for the graduate research assistant position supported through the grant. This grant would fund a graduate student to work with me on the project for 20 hours per week throughout the year and learn about all aspects of research, including project management for a project that includes many interdisciplinary collaborators. Other graduate students who are mentored by me or other faculty involved in the project are also welcome to join the project and receive the same experience as the GRA. For undergraduate students, my college supports undergraduate research through an internship program that provides research experience and course credit. I am currently working (Summer 2019) with an undergraduate research intern on this project. She is learning how to conduct education research and analyze data, and to meaningfully contribute to the project, she is also learning some programming skills. I view undergraduate research interns' responsibilities as half learning about the research topic, methodology, and analysis and half contributing to the project through providing feedback on materials or analyzing data. This program allows me to encourage undergraduates to pursue research and/or computing, and given the demographics of Georgia State students, the undergraduates (and graduates) who I work with are typically from group that are underrepresented in research and computing.

Broader Impact

Because the research and teaching components of the proposed project are intertwined, the applications of the proposed research has immediate broader impact on computational literacy in Georgia, especially in Atlanta. The widespread incorporation of computing education into PST preparation programs is rare, and I believe the Georgia State is the only institution reaching every PST by using computing integration activities in methods classes. Because we are a large university with large cohorts of PSTs every year, we will prepare about 300 PSTs annually for computing integration. Within the years of the proposed project and at Georgia State alone, that is 1500 teachers across all grades and subject areas who can teach computing integrated activities.

In addition to the number of teachers who will learn computing from the project, Georgia State's location and population of students means that at least half of the PSTs are from groups that are underrepresented in computing, including people of color and from low-income families. Most of our teacher graduates will teach in the school districts in which they grew up, meaning that they will be teaching primarily students of color and from low-income families in Atlanta. As a growing technology hub, many Atlanta schools have been given computers from corporate donors, but the teachers have a limited skill set for using the computers. This project will help them use these resources to spread computational literacy for their students.

To ensure that the computing integration activities that PSTs learn at Georgia State will be applicable to their future classroom, we are taking several measures. First, we are using only computing integration activities that are web-based. Therefore, teachers do not need special equipment or kits to implement the activities in their future classroom. They need only a computer, iPad, or even smartphone with an internet connection, which is available for every student (i.e., one-to-one) in most schools in Georgia. Second, we are using computing integration activities in student teaching to ensure that the activities are viable for the classroom and to boost confidence of the PSTs. In the pilot year, enthusiasm for computing integrated activities soared after student teaching because the PSTs saw students' excited to learn about computing, and they successfully implemented the activity. Last, we are directing PSTs towards resource repositories that include many computing integration activities, allowing them to use computing integration in their classroom even if the specific activity that we shared does not fit.

During the proposed project, I will focus on teachers in the metro Atlanta area and explore the research goals deeply within this context. Thorough answers to the research questions will support evidence-based interventions that can be rapidly developed for implementation outside of my institution and my state. Through the rest of my career, I hope to have a national impact on the computational literacy of teachers and their students. I will start building capacity for national implementation throughout the project by building my network through working with my mentors, giving presentations at conferences with a range of audiences, and publishing interdisciplinary journal articles. I will continue working with the Georgia DOE on CS education initiatives to continue to receive feedback on our programs and publicize our work throughout Georgia. Moreover, I will continue to build relationships with national organizations like ISTE and the CSforAll Consortium, in which I am an active research member. This network can improve the dissemination and impact of the excellent work that we are doing at Georgia State to spread computational literacy by sharing the evidence-based interventions with the many organizations who are working towards similar outcomes.

Results from Prior NSF Support

Developing and Assessing Subgoal Labels for Imperative Programming to Improve Student Learning Outcomes (DUE, 1712231; $299,927, 2017-2020). PI: Morrison, B.; Co-PI: Decker, A., Margulieux, L.

Intellectual Merit: This project is identifying subgoals of problem-solving procedures in intro programming courses and creating instructional materials based on subgoal learning to explore long-term effects of subgoal learning on problem-solving performance for students.

Broader Impacts: Build a subgoal-oriented set of worked examples and practice problems for intro CS courses that will help novices develop better foundational knowledge of CS.

Products: Task analysis of subgoals, subgoal-oriented worked example and practice problem pairs, e-book with instructional materials, workshop for intro programming instructors at SIGCSE '19, publications - Decker, Margulieux, & Morrison, 2019 (ICER); Margulieux, Decker, & Morrison, 2019 (ITiCSE); Morrison, Margulieux, & Decker, under review (*Computer Science Education*).