Table of Contents

Rationale for Absolute Priority 1......1
Rationale for Absolute Priority 3......4
Rationale for Competitive Preference Priority......5
Significance: Contributions to increased understanding of effective strategies......6
Significance: New strategies that build on existing ones......8
Project Design......11
Conceptual framework......13
Ensuring feedback and continuous improvement......15
Timeline......17
Qualifications of key personnel and responsibilities......18
Sustainability......19
Dissemination......19
Project Evaluation......20
Implementation Study Design......22
Impact Study Design......23

A Modern Approach to the integration of Programming and Mathematics addresses Absolute Priorities 1 and 3 and the Competitive Preference Priority. *Introduction to Computational Thinking* (**ICT**), currently in use in 15 ninth and tenth grade classrooms in Louisiana, is a course we have developed to teach problem solving, programming and mathematics (algebra and geometry) in a seamlessly integrated way that encourages students' artistic expression and reinforces their sense of STEM self-efficacy. This project will enable us to continue to refine the course, even as we provide initial evidence for its effectiveness.

Rationale for Absolute Priority 1

The ICT course is designed to improve student outcomes in mathematics, informed by an integrated inquiry-based approach to learning mathematical concepts and by an approach to professional development that prepares teachers for project-based learning. In addition, the ICT course aims to teach fundamental principles of computing to students with diverse interests, not only to those aspiring to become software developers.

Integrated learning of math concepts. The design of the ICT course is inspired by principles similar to those guiding the Core-Plus mathematics curriculum: 1) It contains interwoven strands of content from algebra, geometry, probability and discrete mathematics; 2) It has a strong emphasis on modeling; 3) It uses technology to promote reasoning with multiple representations (verbal, numerical, graphical, and symbolic); 4) It focuses on goals in which problem solving based on mathematical thinking (and in our case, also computational thinking) is central; and 5) Teaching materials emphasize active learning, small-group collaboration, and summarizing activities that lead to reflection on the main ideas. A study by Schoen & Hirsch's (2002), which met the WWC standards with reservations, found potentially positive effects of the Core-Plus mathematics curriculum on mathematics achievement for high school students.

Professional development with a focus on inquiry-based learning, high-quality lesson design, a community of learners, and technology integration. Our professional development model promotes the integration of project-based learning and 21st century skills into daily instruction. Participating teachers must go through an intensive five-week all-day summer institute where they learn programming, how to teach programming and how to assess, guide and help students with their learning. Teachers must complete all the student assignments, learn how to review student code and how to assess student work. They work in groups to complete and present their projects, and learn different collaboration strategies, such as pair programming and deliberation on a whiteboard. At the end of the summer institute, most teachers are ready to teach ICT, but they are supported with on-demand individualized coaching and monthly Saturday review sessions. It is also expected that teachers stay active contributors in subsequent years and help mentor teachers new to the program. Some teachers who went through our early phase of PD are starting to develop activities and lessons, for topics such as comparing linear and exponential growth for their regular Algebra I classes, in which they use the coding skills they learned during our PD. This model contains relevant overlap of populations and settings with the eMINTS professional development program. Meyers et al's (2015) review of the eMINTS program, which met the WWC standards without reservations, concluded that students whose teachers had participated in the program were more engaged and had higher math scores than the corresponding control group. We expect teachers participating in our program to provide similar benefits to their students.

Computer Science instruction for all. While there are still no studies meeting the WWC standards that review the effects of teaching computer science to all students, there is wide consensus about its benefits, specifically with respect to computational thinking (Angeli et al, 2016; Buitrago Florez et al, 2017; Denning, 2017; diSessa, 2001; Grover & Pea, 2013; Grover et al., 2015; Pellegrino et al., 2013; Voogt et al., 2015; Weintrop et al, 2016; Wing, 2006, 2008, Yadav et al, 2018). A recent report of the Committee on STEM Education (2018) of the National Science & Technology Council listed computational thinking as one of the three integral elements that should be added to all education. It is also important that citizens understand the role of computing in their lives, and in, particular, the importance of programming literacy (Prensky, 2008; Vee, 2013), which can no longer be thought as the exclusive domain of computer science.

Computational Thinking is routinely exercised by software designers, but it is not only limited to programming. However, beyond an elementary level, illustrating its essence becomes clearer when a high-level programming language is used as the medium (Denning, 2017). Programming plays a major role to enable the demonstration of computational competencies (Grover and Pea, 2013). The ICT curriculum was designed with these criteria in mind: 1) Students taking the course will have a wide spectrum of interests; 2) Most students who take the course are not going to become programmers; 3) Most teachers who teach the course have no computer science background; 4) Despite the above, the course must have rigorous content; 5) Students will demonstrate their learning by creating computer programs; and 6) Creativity and choice within the given constraints is an essential aspect. A question that guided the design of ICT is *What unique elements of programming that students learn in this course could be i) useful later in their life no matter what they do, and ii) difficult to learn in another course?* Answering this question led us to choose a high-level programming language that is based on sound mathematical principles and to ignore low level details of implementation that are important for software professionals but can obscure the universal concepts underlying any computing system.

Rationale for Absolute Priority 3

The ICT course is a **field-initiated innovation** promoting STEM education with a particular focus on Computer Science and Mathematics. The creation of the ICT curriculum was initiated in 2015, as part of a modernization plan by the East Baton Rouge Parish School System (EBRPSS), which partnered with the Louisiana State University (LSU) Gordon A. Cain Center for STEM Literacy to create curricula and academic support structures for the newly created Lee High School. Lee High is a STEM magnet school offering its students three STEM Academies to choose from: Digital Media, Biomedical and Pre-Engineering. Based on a desire to improve student performance in high-stakes math assessment tests (LEAP 2025, 2018), the EBRPSS administrators requested a programming course that would serve, also, to reinforce concepts from Algebra I (9th grade) and Geometry (10th grade). ICT is designed around hands-on activities that help students build mental models of quantitative and formal reasoning that are foundational to mathematical understanding. The ICT course was selected as a core course for all the Lee High academies.

Addressing workforce needs. A report from the Louisiana Workforce Commission (2014) predicted that the "computer and mathematical occupations" will be the ones with the largest ten-year growth of 36.6% by 2022. Similarly, the National Center for Women & Information Technology (2011) projected that up to 77% of future job openings in Louisiana could be filled by people with computing degrees. In spite of the job opportunities, in 2018 only 1.5% of the total number of students taking AP tests in the state took a Computer Science (CS) test, compared with 2.7% nationally.

In 2017, to respond to the workforce needs, the Louisiana Department of Education (LADoE) partnered with the LSU Cain Center to create and pilot new statewide high school STEM graduation pathways, which followed the model pioneered for the three Lee High School Academies and kept the ICT course as a core course in all the Pathways. Schools in Louisiana are able to support this initiative, including their teacher professional development, using LADoE Career and Technical Education (CTE) and Career Development Funds (CDF) ($476 per student per course). In addition, high-school students participating in these courses are eligible to earn silver or gold diploma STEM Endorsements if they complete 4 or 8 STEM Pathway courses, respectively (Louisiana STEM diploma endorsements, 2018). The Pre-Engineering Pathway started a pilot phase at seven schools in the Greater Baton Rouge area in 2017, expanding the number of schools to 31 in 2018, with more schools joining for the 2019-2020 year.

Rationale for Competitive Preference Priority

The ICT curriculum is a hands-on, project-based approach to learning text-based programming using CodeWorld (2019), a Web-based Integrated Development Environment specifically designed for students in eighth to tenth grade. CodeWorld is based on a simplified version of the Haskell language that allows students to create complex drawings with a few commands. CodeWorld has a limited set of graphical primitives to draw circles, rectangles, and text, and to apply translations, rotations, scalings and colors to them. Importantly, the syntax of CodeWorld closely matches mathematical notation, so programming expertise and mathematical competence with functions mutually enhance each other. A mix of guided and free form projects helps students practice fundamental computational concepts, such as abstraction, decomposition and pattern recognition, while simultaneously addressing basic mathematical content in algebra and geometry.

ICT meets the **Competitive Preference Priority** because our project is designed to improve student achievement in computer science, especially for underserved populations. ICT is now being taught in schools from eight districts in the Baton Rouge area, including 3 in EBRPSS. To increase the diversity of the student clientele, we are expanding the implementation to 10 out of 11 high schools in EBRPSS. The district has 41,098 students, 85.5% from underrepresented (URM) minorities (75% African American, 9.1% Hispanic) and 75.5% from economically disadvantaged families. In addition, we will cover four rural districts in Louisiana: Evangeline, Pointe Coupee, Washington and West Feliciana.

Significance

Significance: Contributions to increased understanding of effective strategies

We expect to contribute to increased understanding of two effective strategies.

Strategy 1: PD as accelerated education in a semi-formal setting. An exceptional challenge of PD in Computer Science (CS) is that an overwhelming majority of the teachers--even math and science teachers!--have no background whatsoever in the field. The most expedient way to solve the PD problem for such teachers is to adopt an automated online instructional program for students, and relegate the teacher to a facilitator role of ensuring that students are completing online assignments, and recording their scores. This bypassing of the teacher's instructional role is being widely adopted; however, curricula that can be delivered in this autonomous fashion primarily address procedural competence.

Instead of following this expedient path, we are facing up to the challenge of training teachers, in keeping with the well established idea that student engagement with a competent and knowledgeable teacher is the most influential external factor on students' significant achievement (Hattie, 2003). To address this challenge, we are pursuing a strategy of accelerated instruction for teachers. The accelerated strategy is to focus on fundamental principles while avoiding many technical topics of interest to future software-developers but less relevant for general education. This strategy is possible because the ICT curriculum, itself, is stripped down to conceptual fundamentals. And our second strategy, below, covers the origins and general character of these important fundamental concepts. This project will build knowledge about the feasibility of PD as a means to teach a STEM discipline to teachers new to it. In particular, we will investigate to what extent can a face-to-face five-week all-day summer institute be shortened by moving parts of it online, while keeping it highly effective for novice teachers. This is important in order to better serve remote rural districts.

Our PD efforts in this grant will complement our ongoing work that covers in-service teachers as well as pre-service teachers. Funded by the U.S. DoE Supporting Effective Educator Development (SEED) award *Expanding and Strengthening the STEM Teacher Workforce Through UTeach*, our team is creating two undergraduate courses for undergraduate students who are considering a teaching career in CS: *Computational Thinking for Future Teachers*, and *Computer Science Teaching Methods*.

Strategy 2: Principled approach to teaching computing. Teaching of programming has traditionally been constituted as teaching a particular language (e.g., Python, JavaScript, Java) to the point that students construe the goal as mastering the language rather than engaging with fundamental concepts such as abstraction, automation, computation, data manipulation, data transfer, human-computer interaction, or modeling, to name a few. Denning (2004) introduced a framework for understanding computing in terms of fundamental principles, which summarize current answers to questions such as *What is computation?*, *What is information?* and *What new knowledge can be acquired through computing?* This framework became the seminal work on which the new Computer Science Principles (CSP) AP course was based (College Board, 2015). CSP is based on the idea that programming is a means to an end, and there are different purposes that people pursue when programming, such as solving science and engineering problems, facilitating business practices, creating artistic expressions or producing artifacts for entertainment. While these principles should apply to *any* course on computing, they are commonly seen as if they applied only to CSP, and most computing courses still use a particular language as the central theme of the course.

In the ICT course, learning the language is de-emphasized. Units are organized in themes, such as *Managing Complexity* and *Data and Calculations* that focus on the actual purpose of the programs rather than on the constructs of the particular language used. In this project, we will systematically collect additional evidence on the affordances that instruction based on fundamental principles of computing provides to students who are completely novice to programming, have no initial interest in becoming programmers and are not developmentally ready to take an AP course.

Significance: New strategies that build on existing ones

We move now to discussing two new operational strategies that build on existing ones that are essential for the success of our project: use of a high-level functional language to maximize transfer, and project-based instruction that integrates learning objectives for math and for coding in each project. In addition, there are a few other strategies that are key components of our project, as described in our Logic Model (Appendix G).

Operational Strategy 1: Maximizing transfer. The term Computational Thinking was coined by Papert, whose LOGO language pioneered the idea that learning programming can be useful for learning mathematics (Feurzeig et al., 2011). This idea, though appealing, has been difficult to substantiate empirically (Pea, 1983; Kurland et al., 1986). Recently, interest in the interplay between computing and mathematics has rebounded (Resnick, 2012; Wright et al., 2013). The Bootstrap project (Schanzer et al., 2013, 2015, 2018a, 2018b), in particular, has addressed transfer within a modern conceptual framework. An independent evaluation of the Bootstrap project (McClanahan, 2016) found that *students of commited Bootstrap teachers experienced growth in their knowledge of key algebraic concepts*. While not yet reviewed by WWC, this report provides evidence of the possibility of transfer.

Schanzer et al. (2015) attribute the favorable results of their intervention, in part, to the use of a functional language as the medium. **However**, Bootstrap is not an extensive curriculum, just 17-hour curriculum focused on simple word problems. Our project extends the use of functional programming to a year-long curriculum that covers many different topics in algebra and geometry. We also have replaced the somewhat unnatural syntax of the language used in Bootstrap with a syntax that much more closely resembles and parallels the syntax of algebraic functions.

What makes functional programming so powerful for the teaching of mathematics is the semantic equivalence of functional programming language to a mathematical expression, a fact that is known as the Curry-Howard correspondence (Wadler, 2014). Students creating a program in a functional language are unknowingly also proving a theorem about the relationship between the inputs and the outputs of the associated math problem. That relation becomes explicit in ICT!

Operational Strategy 2: Integrated math/coding projects. Butterfield and Nelson (1989) pointed out that the content of programming needs to be paired with an appropriate teaching methodology. Papert advocated the use of constructionist discovery, hoping that transfer of creative thinking would occur spontaneously. However, Mayer (2004) concluded that guided discovery is actually more effective than unguided discovery, and indeed several studies found evidence of transfer when the teaching approach was more guided (McCoy, 1996; Noss, 1986; Hoyles & Noss, 1987; Subhi, 1999; Clements et al., 2001).

Project-based learning (PBL) is designed around the performance of authentic tasks, in which the creation of a product drives the learning (Prince & Felder, 2006). Barron et al. (1998) described four design principles for PBL: (1) defining learning-appropriate goals that lead to deep understanding, (2) providing scaffolds, (3) providing opportunities for self-assessment and revision, and (4) developing social structures that promote participation and a sense of agency. Whereas the efficacy of PBL has been demonstrated in science and mathematics (Boaler, 2002; Krajcik et al.,1998; Marshall, Petrosino, & Martin, 2010), there is less evidence supporting its impact on computer science. However, CS is naturally oriented towards the creation of software products, which can be assessed by running them and checking whether they work as expected or not. This makes CS a very promising domain for implementation of PBL. In addition, students naturally develop a sense of agency that makes them share their artifacts (and their learning) with other students. It is in the first two design principles above where we extend this strategy beyond what is currently applied. First, we define goals that combine mathematics and programming learning, and then we provide scaffolds that help students accomplish the goals, as described in the Conceptual Framework below.

Here is an example of a small introductory project that nevertheless illustrates our approach. After students learn how to write code to draw a square on the screen, an activity challenges them to draw a square that covers exactly half the area of the output window, which is also square. The students can select the method to draw the square among these: by the position of the center, the length of a side and the angle it is tilted; by the position of the center and the position of one corner; or by the positions of two opposite corners. Most students select the first method (center,side,angle) to create a square whose side length is half the side length of the output window, which produces a square that is too small, with an area just one quarter of the total area. At that point, the teacher can give the students three options: 1) try to figure out which should be the right side length (which is the total side length divided by the square root of two), 2) experiment with different squares, have the program calculate their area, and manually adjust the side length until the area is close to half the total area, or 3) use another method to draw the square. Students who choose method 1 will need to solve an equation and then program that solution in the computer, which will give them a sense of agency when they check that the theoretical solution has actual practical implications. Students who choose method 2 will come to appreciate the advantages of automation to save people from performing tedious and repetitive work. Students who choose method 3 will need to *think outside the box* to realize that a square whose corners are at the midpoints of the sides of the output window can trivially be proven to have an area that is half the total area.

Project Design

Our study will take place in 11 (1st year), 16 (2nd year) and up to 29 schools (last 2 years). A complete list of participating schools including locale code, demographic student and teacher information, and student outcome metrics is included in Appendix G. Two teachers from each school will be trained in ICT and also in Introduction to Engineering Design, which are courses in the LSU Pre-Engineering Pathway. Each school will commit to recruit between 20% and 50% of their 9th grade students for enrollment in the LSU Pre-Engineering Pathway. Among those, half the students will be randomly assigned to ICT and the other half to Intro to Engineering Design. Students not enrolled in the Pathway will comprise the control group. An implementation study and two impact studies will be done. One impact study will compare ICT students with Intro to Engineering students, and the other will compare Pathway students with non-Pathway students. We had to use this two-level QE design because the Pathway courses are electives, and enrollment is not mandatory. The **goals, objectives and outcomes** of our project are displayed in the following table:

Goals	**Objectives**	**Outcomes**
1. Pilot and adapt curriculum materials of the full-year (36 weeks, 180 hours) Introduction to Computational Thinking course (Year 1)	LSU and teachers at high-need schools (EBRPSS Lee and at least one rural school) pilot and adapt the curriculum in an iterative manner focusing on: ● increase project-based learning (PBL) focus ● alignment with math standards of practice ● address computational thinking practices ● culturally relevant pedagogy ● streamline online system	An expert panel of researchers and practitioners review the curriculum and rate it excellent. Interviews and survey results of pilot teachers show satisfaction with final version of curriculum
2. Train and coach teachers to implement ICT curriculum with fidelity, using summer training and year-long coaching	LSU delivers a 5-week teacher training covering: ● programming and computational thinking ● connection with math standards of practice ● pedagogical content knowledge including PBL ● culturally relevant pedagogy ● create a long-term community of practice Year-long coaching: ● Online progress tracking of teacher feedback to students ● Monthly face-to-face and virtual meetings ● Immediate support by answering content and programming questions supplementing online system	Expert panel review PD schedule and training materials and rate it excellent Surveys, interviews and observations of PD training show that all components are integrated All teachers remain engaged during PD and take advantage of year-long coaching
3. Implement ICT course in intended schools	Teachers implement curriculum. Students work on online system activities including: ● PBL activities ● programming and computational thinking ● using math standards of practice Students express themselves in culturally relevant ways	Teachers implement ICT with 2,270 students in a combination of urban, suburban and rural schools all of them serving a majority of high-need students (see Appendix G) Students complete at least 80% of activities Implementation fidelity rubric and observations conclude that curriculum was implemented with at least 75% fidelity.
4. Stakeholder engagement including school administrators, counselors and parents	Information sessions for principals, counselors, parents and students to: ● promote buy-in, ● ensure equitable access by underserved	Enrollment in Pre-Eng. Pathway in each school is at least 20% of the 9th grade students Surveys show at least 80% average on appreciation for CT

	students	& computing for college and career readiness
5. Extend current face-to-face PD model to hybrid delivery	LSU pilots and adapts a hybrid model with online and face-to-face teacher PD ● record and edit videos of training ● pilot in Y3/Y4 and Y5 a hybrid model with at least 4/6/8 teachers ● compare outcomes of teachers attending the hybrid PD and face-to-face PD ● iteratively improved hybrid PD plans and materials	Implementation data shows equivalent results for teachers attending hybrid and regular PD training
6. Improve students' outcomes	Increase student achievement: ● Math and CT practices ● Algebra I ● Geometry Increase student familiarity with PBL instruction	Increased student average math achievement by 10% Increased enrollment in AP courses including CSP, CSA and math by 20% Increased graduation rate by 5%
7. Improve student's attitudinal outcomes	Improve student attitudes towards computing careers	Increase percentage of students with positive attitudes towards computing careers by 20%

Conceptual framework

Our framework has two major components: a Cognitive Model and a Pedagogical Model, which are organized around these principles: keep the cognitive load at all times as low as possible, help students make their thought processes explicit, and build foundations by struggling. Richland et al. (2012) report that international studies reveal that US teachers jump too soon into telling students the solutions without giving them time to come to terms with problems. Students need to be encouraged to work through their frustration, but at the same time, teachers need to hold on longer before they hamper the students' progress.

The **Cognitive Mode**l (see Appendix G) describes the dual role of functions to represent a problem and to specify how to compute its solution. It is organized into 1) a modeling component that describes the quantities and variables of the problem, along with charts or diagrams that represent it, and 2) an algorithmic component that describes the rules, processes and structures as they change over time. Solving a problem requires reasoning about both components at the same time, but each component can be translated into a separate coding construct. Students have difficulty reproducing the higher order thinking involved in science and mathematics because it requires multiple types of reasoning to be performed at the same time, and students' capacity to hold them all in the working memory is exceeded, so they often resort to pseudo-structural understanding (Sfard and Linchevski, 1994). We propose that, in order to reduce their cognitive load, computer code can be used for externalizing thought processes that otherwise would remain implicit. This externalizing of thought calls forth metacognitive processes that can lead to process/object reification, often taken to be the hallmark of abstract thinking (Piaget, 1951, Sfard, 2000). The use of code allows students to reify their fuzzy intuition into something tangible that can be manipulated as they reason about a problem.

The **Pedagogical Model** is built around three main practices:

1) Design a progression from quantitative to symbolic reasoning. We use the design recipe (Felleisen, 2001) to progress from quantitative to symbolic relations by slowly transforming numbers into variables and explicit repetition into automatic loops. Even when operating with numbers, we emphasize the importance of leaving expressions unevaluated, so that students make explicit their quantitative reasoning (Thompson, 1994) and keep the association between symbols and the quantities they represent at all times (Kirshner, 2001). Most experts keep expressions unevaluated for as long as possible because the symbols allow them to exploit symmetries and gain insight about the quantities involved. On the other hand, most novices jump too soon into eager evaluation, because they feel more comfortable working with numbers than with symbols (Torigoe, 2011; Kortemeyer, 2016). Our strategy is designed to build a thoughtful transition from numbers to symbols.

2) Encourage experimenting, but discourage guessing. We define experimenting as an exploration of a problem based on a predetermined strategy to search for a solution, while guessing is the unplanned, repeated generation of candidate solutions. While experimenting helps the student acquire information about the structure of a problem, guessing is a mechanical activity that has little benefit. Students have difficulty distinguishing between both, so an effort should be made to explain the distinction clearly.

3) Connect the code with math/science concepts. Using coding for the purpose of teaching content knowledge is very different from teaching programming as a discipline. When programming is used as an instructional tool the focus is not on efficiency, but on structuring the program so that the parallels between the mathematical or scientific concepts and their representation in the code are highlighted. Code, when seen as a carrier of computational thinking, can also give a more modern perspective to mathematics and science. Our design intends to exploit the structural parallelism between programming and mathematics with as few distractions from the syntax as possible. We have received very positive feedback about the use of Haskell from both teachers and students in our ICT course, who found the language very suitable for this goal due to its similarity with the mathematical language.

Ensuring feedback and continuous improvement

Design Based Research. We will use a Design Based Research methodology (Barab and Squire, 2004, Collins, 1992) to analyze the quality of the project with respect to the following aspects: the educational setting (based on interviews with school administrators), the implementation of the intervention (based on lesson plans and incidence logs), the measurement and data analysis procedures and the iteration process (based on detailed records of all tasks performed and bi-weekly summary reports). We will also systematically collect intermediate measures based on student grades for each activity. A count of the number of students not making enough progress will be used as indicators of the progress of each lesson as it is delivered. This will help us identify risks, such as having lessons that are too difficult, that have gaps in the teaching progression, or are otherwise ineffective. We will also keep track of the difference between the expected time and the actual time to completion of each activity. All these measures will be reviewed regularly by our team and also incorporated in our reports to the external evaluator. We will also try to answer how sustainable the design is after the project ends (based on interviews of participant teachers) and what minimum effort would be necessary to support the teachers (based on calculating the number of person-hours spent).

Iteration Analysis. The data collected by the team will be analyzed in an ongoing manner to allow for the continuous refining of the project. All the researchers will meet every two weeks to discuss assessment feedback, review progress, make any needed modifications and adjust/redesign the intervention with a systematized process of editing and review. The online nature of the intervention will allow for immediate changes and will prevent delays in resources being available to the teachers and students. During the entire project the alterations, reflections of the team, and work products will be archived and analyzed annually to insure progress and patterns of change are being followed with fidelity.

Research Practitioner Partnership (RPP). We intend to form a long-term collaboration between LSU and EBRPSS, Evangeline, Pointe Coupee, Washington and West Feliciana school districts that guides and informs various aspects of design and implementation following the RPP model (Bryk, 2015; Coburn et al., 2013, 2016; Penuel et al., 2015; Gutierrez and Penuel, 2014). Mechanisms to increase trust between the researchers and practitioners include monthly meetings of LSU and school district personnel, and meetings with principals and counselors at each school every semester. Special attention will be given to any concerns from the teachers, principals or counselors, which will be addressed before they become a roadblock to the progress of the project.

Timeline

	Activities
Ongoing	• Biweekly meetings of researchers to review assessments and progress, and make adjustment to the intervention • Monthly meetings of researchers, external evaluators and district administrators • Collect and review documents; surveys of school staff and students, interviews with selected school and program staff, and progress tracking data • Once per year face-to-face meeting of whole team including external evaluator • Visit participating school four times per year: two visits to engage school administrators and parents, two classroom visits • Annual evaluation report • Annual project directors meeting
Year 1 **Oct 2019-** **Jun 2020**	• Post-award meeting in DC • Pilot phase: recruit one pilot rural school in addition to EBRPSS Lee High (300 students) • Refine curriculum, assessment instruments, and PD materials • Refine logic model • Recruit teachers: 22 teachers, at least 8 from rural schools • **Milestone:** Fidelity of implementation rubric completed • **Milestone:** Student online system completed
Year 2 **Jul 2020-** **Jun 2021**	• PD: 22 teachers, at least 8 from rural schools two hours away from LSU. Collect & analyze summer assessments • Year-long follow-up with monthly virtual meetings and undergraduate support • Adjust curriculum, course and PD assessments • Work on online platform to move PD to hybrid model • **Implementation study**: fidelity data collected in 11 schools (~425 students) • **Impact study**: collect administrative data for Study 1 (student-level RCT) & Study 2 (student-level QED) • Recruit 32 teachers, at least 12 from rural schools
Year 3 **July 2021-** **Jun 2022**	• PD: 32 teachers, at least 12 from rural schools. Collect & analyze summer assessments • Year-long follow-up with monthly virtual meetings and undergraduate support • Adjust curriculum, course and PD assessments • Test online PD system • **Implementation study**: fidelity data collected in at least 16 schools (~ 615 students) • **Impact study**: collect administrative data for Study 1 (RCT) & Study 2 (QED) • Recruit 6 new schools:: total of 32 teachers, at least 16 from rural schools
Year 4 **July 2022-** **Jun 2023**	• PD: 32 teachers, at least 16 from rural schools. Collect & analyze summer assessments • Test hybrid model for remote teacher PD; refine software and methods • Year-long follow-up with monthly virtual meetings and undergraduate support • Test optimization of remote support • **Impact study**: collect administrative data for Study 1 & Study 2 • **Replication phase** in 6 new schools • Recruiting 3 new schools: total of 34 teachers, at least 16 from rural schools
Year 5 **Jul 2023-** **Jun 2024**	• PD: 34 teachers, at least 16 from rural schools. Collect & analyze summer assessments • Final test of hybrid model for remote teacher PD; refine software and methods • Year-long follow-up with monthly virtual meetings and undergraduate support • Test optimization of remote support • **Impact study**: collect administrative data for Study 2 (student-level QED) • **Replication phase** in 9 new schools • Recruiting 4 new districts: total of 34 teachers, at least 16 from rural schools

Year 5 **Jul-Sep 2024**	• PD: 34 teachers, at least 16 from rural schools. Collect & analyze summer assessments • **Reporting phase**: complete analysis of data and final report

Qualifications of key personnel and responsibilities

Our interdisciplinary team has worked for four years on the STEM Certification Pathways. **Moreno** (LSU-PI) holds a shared appointment between Physics & Astronomy and the Center for Computation & Technology. **Alegre, Kirshner** and **Neubrander** are members of the LSU Cain Center for STEM Literacy. **Chen** and **Kirshner** are faculty members of the School of Education. **Neubrander** is faculty in the Department of Mathematics. **Necaise** (EBRPSS-PI), **O'Konski** and **Rutledge** are top level administrators. **Navo** is EBRPSS District Grant Writer, and **Underwood**, 2018 EBRPSS Teacher of the Year, is also a Ph.D. student in LSU School of Education. **Neubrander** (Interim Executive Director of the Cain Center) leads the LSU's STEM Certification Pathways. Three members of our team (**Alegre, Moreno** and **Underwood**) have participated since the beginning in the creation of the ICT curriculum, its assessment, and the delivery of the summer teacher training.

This project will be conducted under the direct supervision of the PIs **Moreno** and **Necaise**. They will oversee all aspects of the project, monitor necessary adjustments to the initial plan, handle all logistics, and contact with the external evaluator. The project's data collections will be coordinated by **Navo**, who will be the designated EBRPSS Data Collection Monitor (see Superintendent Drake's letter). **Alegre** will be responsible of completing the development of the ICT curriculum and the online PD platform, and ensure that best computational thinking practices are followed. **Chen** and **O'Konski** will oversee the collection of student assessment and questionnaires, and be responsible for analysing the data and presenting it to the team. **Kirshner** will coordinate the alignment of the curriculum with math standards and the mathematical pedagogical content knowledge. **Neubrander** will be in charge of all communication with school districts and the recruitment of new schools to participate in the project. **Rutledge** will be responsible of the PD pedagogical content knowledge and of reviewing the curriculum to assure best culturally relevant practices. **Underwood** will aid in the collection of teacher participant feedback and help with pedagogical practice issues throughout the project. **Lewis,** EdNW, will serve as our lead external evaluator.

Sustainability

As of 2019, LSU charges $3,940 per teacher for the summer PD and $96 per student during the school year. These fees cover instructors' salaries, maintenance of the online system and support during the school year. The amount charged per student is comparable to other CS providers, such as CodeHS, which is estimated to charge about $2,000 per classroom. Schools in Louisiana can recover their cost through the LaDoE CTE/CDF funds, but some districts do not have the funds to start the program. As more schools join the Pathways, we will be able to reserve some funds for schools that cannot initially afford the costs, such as rural schools where the student enrollment may be low. Currently, most of the cost goes into PD instruction, due to the long period needed to get teachers trained initially. In the last two years of the project, we expect to have our online system ready for moving a large portion of the professional development online, reducing face-to-face presence, and correspondingly the cost, from the current five weeks to two weeks. Due to the nature of the PD, a fully online solution seems unfeasible, because during the 2nd and 3rd week of the PD, most teachers need help to overcome the frustration that learning a difficult subject in a short time span produces. Our success in the past depended on being able to provide support in a holistic way that would be difficult to mimic online.

Dissemination

The curriculum materials, including lessons, teacher guides and assessments, are open source and licensed under the CC-BY license, and they will be available through the Pathways website, as well as linked from the LADoE website. In addition, LSU offers an online system to manage the course. Teachers using LSU's online system can control how the curriculum is delivered to their students. While teachers need to register with LSU to be able to use the system, no personal information is collected from their students, who are not required to register. Automated assessment of some assignments is also provided, but the system does not use any identifiable information from the students other than their submission. While LSU's current focus is in Louisiana, the system is ready to accept requests from districts out of state. We will start announcing the course, along with the results of the evaluation, at teacher conferences (CSTA, ISTE, NCTM, NSTA) as well as publish the research on the project in peer-reviewed publications.

Project Evaluation

Education Northwest (EdNW) will serve as the independent evaluator throughout the 5-year project. Based in Portland, OR, EdNW has more than 50 years of experience in research and evaluation, as well as direct experience evaluating an i3 validation grant in Alaska and a current early-phase EIR grant awarded to Portland Public Schools. The EdNW project team includes researchers with content expertise in STEM teaching and learning as well as methodological expertise in evaluation methods, including designing and conducting implementation fidelity studies and impact evaluations that meet WWC standards.

Evaluation methods

Over the course of the project, EdNW will conduct ongoing formative evaluation to support continuous program improvement, an implementation study to measure implementation fidelity, and an impact study to test the effect on student outcomes. The evaluation questions for each phase of work and data sources are summarized in the table below.

Questions	Data Collection Tools and Measures
Formative evaluation (years 1-4)	
What are the **key elements and activities** of the project and to what extent are they being implemented as intended?	**Documents** (project plans & records, meeting & training materials, feedback forms) **Surveys** (school staff, students) **Interviews** (program administrators, school staff) **Progress tracking data** (project records, classroom observations, attendance records, transcripts)
To what extent did project activities produce expected **outputs**?	
What **barriers to implementation** arose and how were they overcome?	
Implementation study (years 2-3)	
Was the program **implemented with fidelity**? Did implementation fidelity vary across schools?	**Implementation fidelity rubric**
Impact study (years 2-5)	
What effect does the ICT curriculum have on **students' math achievement**? What effect does the Pathways program have on **student accumulation of math and CS credits and likelihood of taking and passing AP math and CS exams**? What effect does the Pathways program have on **students' graduation rate**?	**Administrative data** (demographic files, transcripts, test scores, enrollment records)

Formative Evaluation and Continuous Improvement

Throughout the project EdNW will pursue a mixed-methods evaluation to support the ongoing development and improvement of the program. During the pilot phase, formative evaluation efforts will include a particular emphasis on evaluating the efficacy of the teacher PD model as the program administrators move to a hybrid model that combines face-to-face and online curriculum delivery. Formative evaluation activities will include describing program implementation and stakeholder perceptions of program activities, outputs, and short-term results. Evaluators will use descriptive statistics to analyze quantitative implementation data, such as close-ended survey items and project records and will apply content analysis to make inferences about qualitative data, including training materials, open-ended surveys, interview and observation data.

To help project leaders make real-time adjustments to maximize project benefits, EdNW will provide formative feedback at regular intervals to support iterative refinement of the program model by documenting implementation successes and challenges and lessons learned. After each data collection, EdNW will deliver performance feedback reports with high-level findings and considerations for program improvement. During twice annual in-person meetings, EdNW will facilitate conversations to help the program administrators interpret evaluation findings and plan for upcoming evaluation activities. Finally, EdNW will submit annual reports that summarize program activities and progress on the project goals, using current and longitudinal data. Methods for collecting qualitative and quantitative data are summarized below.

Formative Evaluation Data Sources	
Surveys	Satisfaction surveys will be administered in Years 1-4 to school staff and students to gauge perceptions of the approach and fidelity to the implementation plan.
Interviews	Semi-structured interviews will be conducted with selected school and program staff to understand how implementation is occurring and help identify challenges and successes of the implementation process. Semi-structured interviews will be conducted with at least two project leaders and developers two times a year throughout the project, and once a year with at least two staff members in each participating school in Years 1-3.
Document review	Evaluators will review documents and materials produced in the process of designing and implementing the curriculum and associated training. These data will inform development of survey and interview protocols, and the implementation fidelity rubric.
Progress tracking data	The evaluation team will help program leaders collect and analyze observation and quantitative data to track progress on goals and benchmarks throughout as well as data on student demographics. These data will relate to numbers of participants, which will be tracked through administrative data collection at the sites.

Implementation Study Design

Using what is learned during the pilot phase of the project (Year 1), EdNW will formalize the program logic model to clearly articulate the key project components, mediators, and outcomes (see Appendix G). The refined logic model will then be used to develop a fidelity of implementation rubric by the end of Year 1. The rubric will identify indicators for each key component, data sources for each indicator, and will set measurable thresholds for the implementation of each of the key components of the intervention. Implementation fidelity data will then be collected for all schools participating in the treatment in Years 2 and 3. Fidelity scores will be computed based on how many fidelity indicators were met for each key component and then combined to determine an overall fidelity score (e.g., low, adequate, ideal). The implementation scores will be examined to determine whether varying levels of implementation quality correlate with student outcomes. Implementation study findings will also be used to clarify under what conditions the program works best and to provide feedback on the extent to which it is feasible to replicate the program in other settings.

Impact Study Design

We will conduct two impact studies to test the effects of the curriculum on a comprehensive set of outcomes: Impact Study 1 will be a student-level RCT designed to meet WWC Evidence Standards without reservations to test the effect of the ICT curriculum on near term outcomes; and Impact Study 2 will be a student-level QED designed to meet WWC Evidence Standards with reservations to test the effect of the overarching Pathway curriculum on more distal outcomes. These two studies in tandem will help us understand whether integrating the teaching of computing with mathematics impacts math outcomes as expected, but will also shed light on the broader research question of whether the overall Pathway of courses improves student outcomes in math and CS.

Impact Study 1. It is not possible to randomly assign students to participate in the Pre-Engineering Pathway the ICT curriculum is embedded within. However, once enrolled in the

Pathway it is possible to randomly assign the order in which students take specific courses. This programming feature will allow us to rigorously test the effect of the ICT course on 9th grade math achievement. For this study, students enrolled in the Pathway will be randomly assigned to take ICT in the 9th grade and control students will take an alternative course, Introduction to Engineering Design, that has not been explicitly designed to reinforce math concepts. The control students will then receive ICT in the 10th grade. Random assignment will occur at the student level within schools. This study will include cohorts of 9th grade students entering the program in Year 2, Year 3, and Year 4 for an estimated total of 1,655 students across 16 schools. Although WWC standards do not require establishing baseline equivalence in RCT studies, we will collect baseline information about schools and students as a safeguard in the case of high attrition.

Impact Study 2. All students in the Pathway will be compared to a sample of propensity-score matched peers who did not participate in the Pathway to examine effects on longer-term outcomes of interest including accumulation of credits in CS and advanced math (including AP credits in both subjects), and graduation rates. We will use propensity score methods to match Pathway students to a sample of peers based on CS credit accumulation prior to 9th grade, prior achievement test scores, and demographic characteristics. This study will include the cohort of 9th grade students entering the Pathway in Year 2 and a matched comparison group. Only 20% of students at the participating schools are expected to enroll in the Pathway, thus we expect to have ample untreated students to draw a comparison sample from. We anticipate that this study will include 850 students across 11 schools.

Statistical power. To estimate the size of a treatment effect that the proposed impact studies are powered to detect, we estimated the **minimum detectable effect** (MDE) based on the anticipated parameters for each impact study. Assuming a power level of .8, an alpha of .05, and a two-tailed test, the MDE for Impact Study 1 is .14 and the MDE for Impact Study 2 is .19.

Analysis plan. Because this is a multi-site project, all statistical models will use multilevel regression to estimate the impact of the intervention on student outcomes while also accounting for the nesting of students within schools (and within classrooms if applicable). We will include a full complement of student- and school-level background characteristics (e.g., demographics, size and composition, average achievement) as covariates in analytic models. For Impact Study 1 we will use an intent-to-treat (ITT) approach, where students will be analyzed in the group to which they were randomly assigned. For Impact Study 2 we will use inverse propensity weighting to account for the lack of random assignment to the Pathway, ensuring that the treatment and comparison groups are comparable on observed covariates. We will follow WWC standards to test for baseline equivalence of the analytic sample.

Performance data on outcomes. The main source of data will be administrative records on student outcomes. EdNW will enter into a data-sharing agreement with the districts represented in the study to access student records. We will use extant data to assess math achievement in 9th grade and 10th grade (using state administered standardized math scores from the Louisiana Educational Assessment Program; LEAP) and in 11th grade (using math subtest scores from the statewide administration of the ACT). Student transcripts will be analyzed to examine credit accumulation in CS and math as well as receipt of qualifying scores on AP courses in these domains. We will also analyze graduation rates using enrollment records.

References

C. Angeli, J. Voogt, A. Fluck, M. Webb, M. Cox, J. Malyn-Smith, and J. Zagami. A k-6 computational thinking curriculum framework: implications for teacher knowledge. *Journal of Educational Technology & Society*, 19(3):47, 2016.

S. Barab and K. Squire. Design-based research: Putting a stake in the ground. *The journal of the learning sciences*, 13(1):1–14, 2004.

B. J. Barron, D. L. Schwartz, N. J. Vye, A. Moore, A. Petrosino, L. Zech, and J. D. Bransford. Doing with understanding: Lessons from research on problem-and project-based learning. *Journal of the Learning Sciences*, 7(3-4):271–311, 1998.

J. Boaler. *Experiencing school mathematics: Traditional and reform approaches to teaching and their impact on student learning*. Routledge, 2002.

A. S. Bryk. 2014 AERA distinguished lecture: Accelerating how we learn to improve. *Educational Researcher*, 44(9):467–477, 2015.

F. Buitrago Flórez, R. Casallas, M. Hernández, A. Reyes, S. Restrepo, and G. Danies. Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, page 0034654317710096, 2017.

E. C. Butterfield and G. D. Nelson. Theory and practice of teaching for transfer. *Educational Technology Research and Development*, 37(3):5–38, 1989.

D. H. Clements, M. T. Battista, and J. Sarama. Logo and Geometry. *Journal for Research in Mathematics Education. Monograph*, 10:1–177, 2001.

C. E. Coburn and W. R. Penuel. Research–practice partnerships in education: Outcomes, dynamics, and open questions. *Educational Researcher*, 45(1):48–54, 2016.

C. E. Coburn, W. R. Penuel, and K. E. Geil. Practice Partnerships: A Strategy for Leveraging Research for Educational Improvement in School Districts. *William T. Grant Foundation*, 2013.

Codeworld. An educational environment using haskell. `https://github.com/google/codeworld`, 2019.

College Board. Computer Science Principles: A new, widely available Advanced Placement Course. `http://apcsprinciples.org/`, 2015.

A. Collins. Toward a design science of education. In *New directions in educational technology*, pages 15–22. Springer, 1992.

Committee on STEM Education of the National Science & Technology Council. Charting a Course for Success: America's Strategy for STEM Education. 2018.

P. J. Denning. Great principles in computing curricula. In *ACM SIGCSE Bulletin*, volume 36 (1), pages 336–341. ACM, 2004.

P. J. Denning. Computational thinking in science. *American Scientist*, 105(1):13–17, 2017.

A. A. DiSessa. *Changing minds: Computers, learning, and literacy*. Mit Press, 2001.

M. Felleisen, R. Findler, M. Flatt, and S. Krishnamurthi. *How to Design Programs*. MIT Press, 2001.

W. Feurzeig, S. A. Papert, and B. Lawler. Programming-languages as a conceptual framework for teaching mathematics. *Interactive Learning Environments*, 19(5):487–501, 2011.

S. Grover and R. Pea. Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1):38–43, 2013.

S. Grover, R. Pea, and S. Cooper. Designing for deeper learning in a blended computer science course for middle school students. *Computer Science Education*, 25(2):199–237, 2015.

K. D. Gutiérrez and W. R. Penuel. Relevance to practice as a criterion for rigor. *Educational Researcher*, 43(1):19–23, 2014.

J. Hattie. Teachers Make a Difference, What is the research evidence? 2003.

C. Hoyles and R. Noss. Synthesizing mathematical conceptions and their formalization through the construction of a logo-based school mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 18(4):581–595, 1987.

D. Kirshner. The structural algebra option revisited. In *Perspectives on school algebra*, pages 83–98. Springer, 2001.

G. Kortemeyer. The losing battle against plug-and-chug. *The Physics Teacher*, 54(1):14–17, 2016.

J. Krajcik, P. C. Blumenfeld, R. W. Marx, K. M. Bass, J. Fredricks, and E. Soloway. Inquiry in project-based science classrooms: Initial attempts by middle school students. *Journal of the Learning Sciences*, 7(3-4):313–350, 1998.

D. M. Kurland, R. D. Pea, C. Clement, and R. Mawby. A study of the development of programming ability and thinking skills in high school students. *Journal of Educational Computing Research*, 2(4):429–458, 1986.

Louisiana Educational Assessment Program (LEAP 2025). Louisiana Department of Education High School Assessments. `http://www.louisianabelieves.com/measuringresults/assessments-for-high-schools`.

Louisiana STEM diploma endorsements, 2018. Louisiana Department of Education. `http://bese.louisiana.gov/docs/rulemaking-docket/summary-report-december-2018-noi-re-bulletin-741-section-2317.pdf?sfvrsn=2`.

Louisiana Workforce Commission. Employment forecasts by occupation and industry. workforce investment council, 2014. `http://www.laworks.net/Downloads/PR/WIC/WICPresentation20140610.pdf`.

M. T. Marshall JA, Petrosino AJ. Preservice teachers' conceptions and enactments of project-based instruction. *Journal of Science Education and Technology*, 19(4):370–86, 2001.

R. E. Mayer. Should there be a three-strikes rule against pure discovery learning? *American Psychologist*, 59(1):14, 2004.

McClanahan, W. S. and Pepper, S. K. and Polin, M. "I program my own videogames": An Evaluation of Bootstrap. `https://www.bootstrapworld.org/impact/EvalReportDec2015Medium.pdf`, 2016.

L. P. McCoy. Computer-based mathematics learning. *Journal of Research on Computing in Education*, 28(4):438, 1996. ISSN 08886504.

C. V. Meyers, A. Molefe, W. C. Brandt, B. Zhu, and S. Dhillon. Impact results of the eMINTS professional development validation study. *Educational Evaluation and Policy Analysis*, 38(3):455–476, 2016.

National Center for Women & Information Technology. *Computing Education and Future Jobs: A Look at National, State and Congressional District Data*, 2011. `https://www.ncwit.org/edjobsmap`.

R. Noss. Constructing a conceptual framework for elementary algebra through logo programming. *Educational Studies in Mathematics*, 17(4):335–357, 1986.

R. D. Pea. Logo Programming and Problem Solving. [Technical Report No. 12], 1983.

J. W. Pellegrino, M. L. Hilton, et al. *Education for life and work: Developing transferable knowledge and skills in the 21st century*. National Academies Press, 2013.

W. R. Penuel, A.-R. Allen, C. E. Coburn, and C. Farrell. Conceptualizing research–practice partnerships as joint work at boundaries. *Journal of Education for Students Placed at Risk (JESPAR)*, 20(1-2):182–197, 2015.

J. Piaget. *The child's conception of the world*. Number 213. Rowman & Littlefield, 1951.

M. Prensky. Programming is the new literacy. *Edutopia magazine*, 2008.

M. J. Prince and R. M. Felder. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of engineering education*, 95(2):123–138, 2006.

M. Resnick. Point of view: Reviving papert's dream. *Educational technology: The magazine for managers of change in education*, 52(4):42–45, 2012.

L. E. Richland, J. W. Stigler, and K. J. Holyoak. Teaching the conceptual structure of mathematics. *Educational Psychologist*, 47(3):189–203, 2012.

E. Schanzer, K. Fisler, and S. Krishnamurthi. *Bootstrap: Going Beyond Programming in After-School Computer Science*. SPLASH-E (Education track of the OOPSLA/SPLASH conference), 2013.

E. Schanzer, K. Fisler, S. Krishnamurthi, and M. Felleisen. Transferring Skills at Solving Word Problems from Computing to Algebra Through Bootstrap. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 616–621. ACM, 2015.

E. Schanzer, K. Fisler, and S. Krishnamurthi. Assessing bootstrap: Algebra students on scaffolded and unscaffolded word problems. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 8–13. ACM, 2018a.

E. Schanzer, S. Krishnamurthi, and K. Fisler. Creativity, customization, and ownership: Game design in bootstrap: Algebra. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, pages 161–166. ACM, 2018b.

H. L. Schoen and C. R. Hirsch. The Core-Plus mathematics project: Perspectives and student achievement. In *Standards-based school mathematics curricula: What are they? What do students learn?, In S. Senk & D. Thompson (Eds.)*, pages 311–343. Hillsdale, NJ: Lawrence Erlbaum Associates, 2002.

A. Sfard. Symbolizing mathematical reality into being–or how mathematical discourse and mathematical objects create each other. *Symbolizing and communicating in mathematics classrooms: Perspectives on discourse, tools, and instructional design*, pages 37–98, 2000.

A. Sfard and L. Linchevski. The gains and the pitfalls of reification-the case of algebra. In *Learning mathematics*, pages 87–124. Springer, 1994.

T. Subhi. The impact of logo on gifted children's achievement and creativity. *Journal of Computer Assisted Learning*, 15(2):98–108, 1999.

P. W. Thompson. The development of the concept of speed and its relationship to concepts of rate. *The development of multiplicative reasoning in the learning of mathematics*, pages 179–234, 1994.

E. Torigoe. How numbers help students solve physics problems. *arXiv preprint arXiv:1112.3229*, 2011.

A. Vee. Understanding computer programming as a literacy. *Literacy in Composition Studies*, 1(2), 2013.

J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav. Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4):715–728, 2015.

P. Wadler. Church's coincidences. In *Keynote SICSA PhD Conference*, 2012.

D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky. Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25(1):127–147, 2016.

J. M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

J. M. Wing. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881): 3717–3725, 2008.

G. Wright, P. Rich, and R. Lee. The influence of teaching programming on learning mathematics. In R. McBride and M. Searson, editors, *Society for Information Technology & Teacher Education International Conference*, volume 2013, pages 4612–4615, 2013.

A. Yadav, P. Sands, J. Good, and A. Lishinki. Computer Science and Computational Thinking in the Curriculum: Research and Practice. *Second Handbook of Information Technology in Primary and Secondary Education*, pages 89–106, 2018.