Integrating Computer Science and Computational Thinking into Elementary Science: Lessons Learned from the Maker Partnership

By Cheri Fancsali, Zitsi Mirakhur, Sarah Klevan, and Edgar Rivera-Cash

January 2022
Acknowledgements

The Maker Partnership was supported through a grant from the National Science Foundation (Award #1742320). This report represents the work of many individuals in addition to the authors. At the Research Alliance, Ethan Crasto provided analytic support; and James Kemple, Alexandra Adair, Kathryn Hill, Chelsea Farley, and Dariana Almeyda-Vega reviewed and provided feedback on drafts. Chelsea Farley edited the report, and Dariana Almeyda-Vega provided copyediting and production support. Roger Horton of Schools That Can and Stephen Gilman of MakerState also reviewed and provided feedback on this report, and contributed to the development of the recommendations.

We are grateful to our practice-side partners at Schools That Can—especially Roger Horton and Casey Lamb—and at MakerState—especially Stephen Gilman, Adam November, and Naved Khan—for their wisdom, commitment, and hard work in this partnership. We also deeply appreciate the contributions of our project’s external advisory group: Bronwyn Bevan, Leigh Ann Delyser, Okhee Lee, and David Wells. They were invaluable thought partners and provided expert guidance and advice. Finally, we are grateful to all of the educators and students who participated in this work. They were generous with their time and in sharing their experiences with us.

Leadership Team

James Kemple
Executive Director

Cheri Fancsali
Deputy Director

Chelsea Farley
Communications Director

About the Research Alliance

Housed at NYU Steinhardt, the Research Alliance for New York City Schools is an independent, nonpartisan research center that conducts rigorous studies on topics that matter to the City’s public schools. We strive to advance excellence and equity in education by providing evidence about the policies and practices that promote students’ development and academic success.

Steering Committee

Augusta Souza Kappner, Chair
President Emeritus, Bank Street College of Education

Jennifer Jones Austin
Chief Executive Officer & Executive Director, FPWA

Mark Cannizzaro
President, Council of School Supervisors and Administrators

Mark Dunetz
President, New Visions for Public Schools

Katherine Fleming
Provost, New York University

Frances Lucerna
Co-Founder and Executive Director, El Puente

Christine Mangino
President, Queensborough Community College

Félix Matos Rodríguez
Chancellor, The City University of New York

James Merriman
Chief Executive Officer, New York City Charter School Center

Michael Mulgrew
President, United Federation of Teachers

David Banks
Chancellor, New York City Department of Education

Michelle Yanche
Executive Director, Good Shepherd Services

Research Alliance publications are supported by a small group of funders who underwrite our core operations, including research capacity, communications, and public engagement efforts. These funders include Carnegie Corporation of New York, the Catherine and Joseph Aresty Foundation, the New York Community Trust, the Wallace Foundation, and the William T. Grant Foundation.

Our publications reflect the findings, interpretations, and conclusions of the Research Alliance and not necessarily those of our funders or individual Steering Committee members.
Integrating Computer Science and Computational Thinking into Elementary Science: Lessons Learned from the Maker Partnership

Executive Summary

The perspective that computing and computational thinking (CT) are necessary competencies for the 21st century is increasingly pervasive (NASEM, 2021). Computational concepts and methods—problem solving, designing systems, refining the steps in a process, and tinkering toward creative solutions—are relevant in nearly every discipline, profession, and industry (Grover & Pea, 2018). As a result, there has been a push to increase access to and participation in computer science (CS) instruction throughout the K-12 curriculum, with widespread efforts to integrate CS/CT across subject areas (Barr & Stephenson, 2011; Grover & Pea, 2018). However, evidence suggests that teachers often lack the capacity to provide students with high-quality learning experiences that integrate CS and CT into their courses (Goode, Margolis, & Chapman, 2014; Gallup, 2015). Further, there is a lack of research on the training and support that teachers need to play this role effectively (Cooper et. al., 2016; NASEM, 2011; Yadav, Stephenson & Hong, 2017).

To address these problems, the Research Alliance for New York City Schools, MakerState, and Schools That Can established a research-practice partnership called the Maker Partnership Program (MPP). Beginning in 2016, MPP developed and tested a model for building individual teacher capacity to integrate CS/CT into elementary science classes using Maker pedagogy. The Maker approach is based on the principles and practices of the engineering design process—an iterative cycle consisting of defining a problem; researching, planning, prototyping and testing solutions; and refining the solution (NRC, 2012). The hands-on, interdisciplinary nature of Maker activities makes them promising for integrating CS/CT into science content and broadening CS participation of those who have historically been underrepresented (Bevan, 2017; Castek et al. 2019; Halverson & Sheridan, 2014).

The following pages highlight key findings from MPP. Drawing on multiple data sources, our study produced lessons about building teachers’ capacity to integrate CS/CT into science through Maker pedagogy, the facilitators and challenges to integrating CS/CT into science instruction, and students’ experiences in such learning environments. Below, we briefly describe these lessons and share related recommendations for professional development (PD) providers, school leaders, and teachers who are interested in engaging in similar work. The insights and recommendations we highlight emerged from the collective sense-making of the research findings that we engaged in through our research-practice partnership.
What We Learned About Building Teachers’ Capacity to Integrate CS/CT into Science

Developing the Capacity of Novice Teachers to Integrate CS/CT

Most MPP teachers integrated lessons by asking students to use Scratch—a block-based programming language—to create a simulation or model of a natural phenomenon (such as the water cycle, energy conversions, or the movement of the earth and sun), which demonstrated students’ understanding of the phenomena they were studying. Teachers may need additional PD and support to make use of more sophisticated CS skills and tools. (This might include, for example, modeling complex phenomena, or using CS to analyze, organize, and display data.)

Nonetheless, teachers with little or no prior experience in CS were able to learn CT concepts and basic to intermediate Scratch programming, and feel comfortable implementing what they learned into their instructional practices, within a fairly short period of time (i.e., over the course of two school years). Notably, we found that teachers needed some facility, but not advanced Scratch skills, to use it as a tool for integrating CS/CT into their science lessons. This bodes well for efforts that are attempting to integrate CS/CT for elementary students at a large scale.

Types of PD and Support That Were Most Helpful

Teachers benefited from sustained engagement in PD, with sessions spread throughout the year. This allowed them to learn new concepts, approaches, and skills in a session, try them out with their students, and then come together again to review their successes and challenges and get more feedback and support. It also allowed teachers to provide formative feedback to our practice partners, which helped improve the PD and support being provided.

Teachers found a number of characteristics of the in-person PD to be particularly effective. These included hands-on learning (e.g., teachers learned CS/CT concepts and skills by creating their own Scratch programs), modeling of lessons and pedagogical approaches, experiencing the lessons as students would, guided practice in using new CS/CT skills, and time for planning and collaboration with other teachers.

MPP coaches provided “wraparound” supports (such as site visits to conduct classroom observations, provide feedback, and meet with teachers and school leaders; emails; and conference calls), which supplemented the knowledge and skills that teachers learned during PD sessions. We found that these activities enhanced coaches’ understanding of the contexts in which teachers were working, making it possible to provide better, more customized support.
Factors That Affected Implementation

There is no "one size fits all" model for who should be responsible for integrating CS/CT into elementary level science (core teachers, science teachers, CS/tech teachers), or where it should occur (in a science class or in a CS/tech class). Rather, integration can occur in a variety of ways and must reflect the specific context, resources, and needs of the school.

Providing resources and materials (e.g., lesson plans and units, model projects and skill-building videos for students, student assessment rubrics) helped teachers integrate CS/CT into science instruction and saved them time and effort. However, even with access to high-quality resources, teachers needed to modify or adapt most lessons to differentiate supports and align activities with the specific science topics they planned to teach. Teachers benefitted from structured time for this work during MPP PD sessions—as well as the guidance and support of MPP coaches and teacher colleagues.

Significant logistical issues related to schools' technology infrastructures (e.g., hardware and internet access, setting up Scratch accounts and getting students logged in, etc.) had to be addressed for successful integration of CS/CT into science. Teachers benefitted from specific support and assistance troubleshooting and establishing routines and procedures to mitigate these issues.

Teachers valued opportunities to collaborate with peers to design lessons, share pedagogical and instructional practices, provide and receive feedback, troubleshoot, and share materials and resources. Working in a school that has an overall vision for CS/CT and concurrent CS/CT efforts (e.g., the districtwide CS4All initiative) seemed to support implementation of MPP by creating an environment where teachers have a community of peers with whom they can share resources and collaborate.

Non-science teachers in particular (e.g., CS or technology cluster teachers) benefitted from collaborating with science teachers to support integration. MPP's PD and supports focused on building skills and knowledge about CS/CT, Maker pedagogy, and integration across disciplines. It presumed that teachers had science content expertise and experience teaching science. However, this was not always the case. Non-science teachers addressed this gap by working closely with colleagues who could provide guidance on the science content.

Engaging school leaders in MPP was challenging and time consuming, but proved to be an essential element in successful implementation and sustainability. School leaders are in a position to support teachers in overcoming common barriers, such as facilitating class scheduling to allow for integration of CS/CT, carving out planning time with other teachers integrating CS/CT, allowing teachers release time so they can attend PD, and ensuring that teachers have access to adequate internet, hardware, software, and other needed resources.

The Value-Add of a Maker Approach

Using a Maker approach provided teachers with a common language and consistent framework for planning and delivering instruction that facilitated student engagement,
creativity, collaboration, persistence, reflection, and independence. This framework helped teachers’ shift their pedagogical approach from instructors transferring knowledge and skill to facilitators of learning. Structuring lessons around the Design Cycle (described below) allowed students to practice the steps that scientists take in exploring natural phenomena (e.g., gathering information and brainstorming solutions, creating prototypes or models of the solution, testing out and improving the design), and in doing so, facilitated the use of computational thinking. Finally, teachers found Maker pedagogy to be an effective approach for integrating CS/CT into science because it allows multiple entry points for students with a wide range of CS/CT skill levels.

Student Experiences

Building CS/CT competencies. Integrating CS/CT into science creates opportunities for students to build fundamental CS skills (e.g., how to create simple programs using motion, control, and event blocks). Similar to teachers, students were able to fairly quickly learn block-based programming, such as Scratch, through scaffolded and guided instruction. Peer collaboration (a key Maker pedagogy practice) seemed to facilitate students’ rapid uptake of coding skills and helped teachers address a wide range of CS abilities in their classes.

At the same time, MPP students spanned the spectrum in terms of CS skills, with many having little or no prior CS experience. This raises important questions about when and how to provide students with the support they need to learn the foundational CS skills that will allow them to use CS/CT to explore scientific phenomena. Science teachers may find it difficult to carve out time from already limited periods of instruction to build these core CS skills. Possible alternatives include collaborating with computer science or technology teachers to provide CS instruction to students, or making basic CS skills a prerequisite to an integrated class.

Outcomes for students. Teachers observed that MPP activities were engaging, encouraged and provided multiple opportunities for peer collaboration and feedback, and improved problem-solving skills, suggesting promising outcomes for students. Our ability to directly measure student learning and attitudes was hampered by the abrupt transition to remote learning as a result of the COVID-19 pandemic. Additional research is needed to fully understand if and how CS/CT integrated into science instruction through a Maker approach improves student learning in either content area.

Recommendations

Emerging from our research and the experience and insight of the MPP partners, we offer several recommendations for PD providers, school leaders and teachers interested in engaging in similar work.

For PD Providers:

- Collect quick turn-around data on teachers’ development of CS/CT skills and pedagogical practices so that PD sessions can better meet their needs.
• Supplement in-person PD with support between PD sessions (e.g., phone calls, emails) to address teachers’ specific needs and questions as they attempt to implement what they have learned.

• Provide teachers with high-quality lesson plans and resources that are aligned with the school’s science curriculum (and that they can easily modify); this is especially important when teachers have little or no prior experience with CS.

• Prepare teachers to address logistical challenges, such as getting students set up on equipment, logging into computers and Scratch, and storing and maintaining the equipment.

• To the extent possible, work with school leaders and teachers to schedule all PD sessions for the year before the school year starts. This helps leaders and teachers avoid conflicts and preserve the time needed to attend. In addition, consider a mix of shorter, more frequent sessions and online sessions that may be less burdensome for teachers to attend.

• Provide school leaders with guidance about how to support teachers’ development of integration and Maker practices. For example, coaches could conduct observations with school leaders and provide a rubric they can use to offer feedback to teachers.

**For School Leaders:**

• Facilitate scheduling, common planning time, and collaboration with other teachers, and address the infrastructure needs that are common barriers to integrating CS/CT into science. Teachers need these resources before and during the school year.

• Integration is inherently multidisciplinary. Consider including both science and CS/technology teachers from the start, in a collaborative effort to help make the best use of staff knowledge and resources and create a foundation for scaling within the school.

• Consider implementing efforts such as MPP with multiple teachers to facilitate collaboration and sharing of resources. Align and collaborate with other CS-related efforts in the building to foster a community of practice.

• Provide teachers with ample time to revise and adapt lessons and activities to address their specific contexts (e.g., to differentiate supports for students, accommodate special needs, and meet students where they are in terms of prior CS experience, and to align lessons with the school/district science curriculum).

• Provide teachers with opportunities to try out new pedagogical approaches and lessons in a lower-stakes environment (e.g., absent of high-stakes teacher evaluations), such as afterschool programs.

**For Teachers:**

• Use existing MPP lessons and lesson templates (available in the Resources section of this report) as a launch pad for adapting and developing your own integrated
lessons that take into consideration the needs, skills, and interests of your students and alignment to the classroom science curriculum.

- Anticipate and try to address logistical and technical issues (such as having equipment in the classroom, charged and ready to use; providing students laminated login instruction cards to assist with logging into computers and Scratch; testing internet connections and software). Avoid cutting into instructional time by tackling these issues before students arrive for class.

- Work with school leaders to schedule common planning periods or times to collaborate with other teachers to share support, information, and resources.

- Build in “unstructured” coding time to allow students who are ready to go farther in their explorations to do so, and to allow students to express their creativity. This also provides opportunities for teachers to work individually with students who need extra support.

- Use peer collaboration so that students can help each other learn (particularly programming skills). Peer collaboration helps students iterate and improve their own work and develop independence from the teacher, and allows the teacher more time to work one-on-one with students who can benefit from additional support.

Next Steps
Building on the lessons from MPP presented in this report, we aim to expand our research-practice partnership to address new problems of practice. Specifically, we are planning to design and test a schoolwide approach to sustainably integrating CS/CT. This work will address the pressing need for models that effectively engage underrepresented students in CS and will continue to inform efforts at CS/CT integration more broadly.

MPP Publications and Resources
In addition to this report, we have published two other papers on MPP. The first paper, “Making” Science Relevant for the 21st Century: Early Lessons from a Research-Practice Partnership, published as part of the FabLearn conference proceedings, describes early findings from Year 1 of project implementation. In a second paper, Making Research Practice Partnerships Work: An Assessment of The Maker Partnership, published as part of the Research in Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) conference proceedings, we describe our assessment of the health of our RPP, and lessons learned.

Resources created for this project can be found on the MakerState website. We encourage others engaged in similar work to use these resources and adapt them as needed to support their practice.
Introduction

Since President Obama’s launch of the Computer Science for All (CS for All) initiative in 2016, there has been a surge of activity aimed at bringing computer science (CS) to all students, with a particular focus on students who have been historically underrepresented in the CS field. This activity is motivated by wide acknowledgement that students need access to instruction in CS if they are to meet the challenges of the 21st Century.

But what does CS entail? The Computer Science Teachers Association defines CS as the “study of computers and algorithmic processes” and asserts that learning CS involves understanding key concepts (e.g., computing systems, networks, impacts of computing, etc.), as well as applying specific practices. These practices include recognizing and defining computational problems, developing and using abstractions, creating computational artifacts (including but not limited to coding programs), and communicating about computing. When one implements these practices, they are carrying out computational thinking (CT) or “the thought processes involved in expressing solutions as computational steps” (CSTA, 2020). We argue that practicing CS involves thinking computationally, which is why, in this report, we refer to the two together, as “CS/CT.”

Unfortunately, it is not yet clear how teachers might foster CS/CT skills among elementary-aged students. While some recent guidance suggests that instruction should begin as early as kindergarten, with opportunities to apply CS/CT across subject areas (Barr & Stephenson, 2011; Grover & Pea, 2018), most teachers do not have the needed background knowledge and skills. School leaders cite the lack of teachers with CS/CT content and pedagogical skills as a key factor limiting their ability to offer instruction in this area (Goode, Margolis, & Chapman, 2014; Gallup, 2015; Lee, Dombroski, Angel, 2017).

There is also a lack of information about how to effectively integrate CS/CT into other academic subjects. Efforts to offer high-quality CS integrated into other disciplines have been hampered by a dearth of research on exactly how to do it, including research that would shed light on the training and support teachers need to make it happen (Cooper et al. 2016; Sands, Yadav & Good, 2018; Yadav, Stephenson, & Hong, 2017). Moreover, it is challenging for teachers to focus on integrating CS/CT in light of the many competing demands on their time. Some research indicates that teachers may not prioritize CS/CT because they don’t feel they can spend class time on instruction that is not directly tied to standardized test preparation (Gallup, 2015).

Despite these challenges, there remains strong interest in integrating CS/CT into other subject areas, particularly at the elementary-level. This interest has been bolstered by evidence that the integration of CS/CT into content areas such as science not only increases students’ CS/CT skills, but also supports deeper learning of the content (e.g., Blikstein & Wilensky, 2008; Peel, Sadler, & Friedrichsen, 2019). CS/CT integration also equips students with the skills to become creators, not just consumers of technology (Smith, 2016), to use CS/CT as a tool for social justice, and to solve real-world problems (Mirakhur, Fancsali & Hill, 2021).
Our research-practice partnership (RPP), the Maker Partnership Program (MPP), was created to meet the challenges and seize the opportunity outlined above. Our goal was to design and test a model for building teacher capacity to integrate CS/CT into elementary science classes using a Maker approach. The Maker approach is based on the engineering design process, with students brainstorming and developing solutions, using technology to create prototypes, and then testing and refining those prototypes together. Our RPP developed lessons and activities that integrate CS/CT into science; facilitated professional development to help teachers learn CS/CT and Maker pedagogy; and supported teachers as they implemented what they learned into their science instruction.

We designed the MPP PD and supports to be implemented over two school years: 2018-2019 and 2019-2020. In the 2018-2019 school year, teachers implemented the MPP curriculum activities in an afterschool setting (see the “Year 1 of MPP” textbox for more information). In the 2019-2020 school year, teachers implemented the curriculum activities as part of their in-school science instruction. This design allowed teachers to test out and get more comfortable with CS/CT activities in a lower-stakes, more flexible afterschool environment before integrating them into a formal classroom setting. It also allowed us the opportunity to test out and improve different types of support, PD facilitation styles, and CS/CT integrated lessons plans and units over the two years. As in schools across the world, the switch to remote learning in the spring of 2020 as a result of the COVID-19 pandemic profoundly affected our work, which we explain in more detail below.

This report describes key findings from the project, with a focus on Year 2, when teachers integrated CS/CT into their in-school science instruction. It is organized into three sections. In the first section, we describe the core components of the MPP project, including the MPP instructional model, the context within which it was implemented, the RPP partners and how we worked together, and the research questions that drove our work. Next, we describe findings from the research we conducted through the partnership, including the frequency with which teachers integrated CS/CT into their science instruction and implemented a Maker approach; adaptations they made to MPP lessons; facilitators and barriers to implementation; outcomes for teachers; and outcomes for students. Throughout this section, we provide short vignettes and examples of integrated instruction and PD based on our observations. In the final section, we provide a summary and discussion of lessons learned through this work and recommendations for PD providers, school leaders, and teachers. Resources created and adapted for MPP that may be helpful to others engaged in similar work are located at the end of this report.
Year 1 of MPP

In the first year of our program (2018-2019), teachers were expected to implement a Maker approach with CS/CT in afterschool activities at their school. The aim was to develop teachers’ CS/CT skills and give them the opportunity to facilitate students’ learning in a low-stakes environment. Pairs of teachers at each of our participating schools co-taught the following units:

- **People programming:** Over the first 5 to 7 weeks, teachers introduced students to CS/CT concepts and skills using physical movement and art-based cooperative learning activities.
- **Scratch basics:** Teachers spent the next 6 to 8 weeks covering the basics of programming in Scratch, including if/then statements, movement of sprites, and setting backdrops.
- **Ozobots:** Students spent about 5 weeks learning to program small robots, building on the lessons they learned in the previous units. CS skills included learning about conditional statements and loops.
- **Scratch science:** This 11- to 13-week unit was most similar to what teachers did during the second year of MPP. In this unit, students used Scratch to create simulations of natural phenomena drawn from their science classes (e.g., sunrise and sunset, making waves, and food/energy).

Teachers met with their students after school once a week for 90 minutes. The curriculum was centered on “challenges” students had to complete (e.g., designing a game using Scratch), which were aimed at teaching the focal CS/CT skills for that unit.

Our interview and survey data suggest that, over the first year of MPP:

- Teachers responded positively to the PD and the support they received through MPP. Teachers found the in-person PDs, coaching, and online resources helpful to their development.
- Students collaborated with one another as they worked on projects together. They were less likely to agree with the survey statement, “I do not like when people suggest changes to my work” at the end of the year relative to the beginning.
- Student confidence, particularly in their CS skills, increased over the course of the year. Students were more likely to agree with survey statements about their programming ability (“I can make a computer program”), and rated their skills at programming (“I am good at computer programming”) higher at the end of the year than the beginning.

The first year of MPP yielded useful lessons that we continued to build on in the second year. This included:

- **Active, hands-on PD:** The MPP team continued to put teachers in their students’ shoes during PDs, so they could experience a model of effective lesson delivery while learning the content.
- **Creating a community of practice among teachers:** Coming together several times a year helped teachers get to know each other better—and use one another as resources.
- **Distributing PD over the course of the year so teachers could continue to digest content, try out lessons or pedagogical techniques, and receive additional support.**

See the paper, *“Making” Science Relevant for the 21st Century: Early Lessons from a Research-Practice Partnership*, for other findings from Year 1 of MPP.
The Maker Partnership Program

In this section, we describe the components of the Maker Partnership Program, including how CS/CT was integrated into science instruction through Maker pedagogy, how the partners of the RPP worked together, and the research we pursued.

The MPP Instructional Model

The MPP instructional model centered around two key features that work together: the integration of CS/CT into science and the implementation of a Maker approach. Because a Maker approach naturally draws on interdisciplinary practices, such as aspects of the engineering design process, scientific inquiry, math, and technology (Kafai & Peppler, 2010; Peppler, 2013; Sheridan, et. Al., 2014), it is well positioned to integrate CS/CT and science. Below we describe how these key components were infused into science instruction through MPP.

Integrating CS/CT into Science

MPP is premised, in part, on the idea that using CS practices, tools, and technologies can deepen students’ understanding of the scientific phenomena covered in elementary science curriculum. Teachers addressed a variety of science topics in their integrated lessons, including electrical energy, the solar system, energy conversion, motion, food chains and the ecosystem, and animal traits. The CS tools teachers used to explore these topics included “unplugged” activities (e.g., hands-on activities such as games and puzzles that teach CS/CT concepts without computing devices or technology), Scratch (a block-based programming language and online community), and Ozobots (small programmable robots). For example, students used Scratch to build a program that illustrated the water cycle and its effect on the environment; a model of the sun in the sky that demonstrated the change in daylight as the earth rotates; and a game that demonstrated how the forces of acceleration and drag affect the velocity of a moving object, such as a spaceship.

At the same time that students were learning scientific content, MPP activities and lessons aimed to build their knowledge and skill around CS/CT concepts and practices. This included basic coding skills and an understanding of foundational coding concepts such as loops, conditionals, and variables. Practices developed through MPP activities included abstraction (e.g., breaking down ideas or concepts to generalize or make it easier to understand), algorithmic thinking (e.g., identifying the sequence of steps that need to be taken to achieve a goal), working with data, and creating models that can be used to simulate how natural phenomena occur. See Table 1 below for definitions of the key CS and CT terms used in this report.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
</table>
| Abstraction              | ● Reduce complexity by focusing on the main idea.  
                              ● Filter out or hide details we don’t need in order to focus on the problem.                                                            |
| Decomposition            | ● Take a big idea/phenomenon and break it down into smaller parts for the purpose of modeling.  
                              ● Pull out essential elements of a problem to build a representative model.                                                              |
| Pattern Recognition      | ● Identify similarities or differences among groups of things.                                                                           |
| Modeling and Simulation  | ● Create models and simulations to represent and analyze a process, system, or natural phenomenon.  
                              ● Design a simulation to test, analyze, and/or evaluate a model.                                                                           |
| Algorithms and Programming| ● An algorithm is a step-by-step set of directions to accomplish a goal. Programming is the process of converting an algorithm into a particular language that can be read by a computer. |
| Working with Data        | ● Incorporate data (create, modify or manipulate) into the analysis of a natural phenomenon.                                                 |
| Iteration                | ● Test and revise a prototype to improve the design or solve a problem.                                                                      |
| Playtesting              | ● Students test one another’s prototypes and provide feedback.                                                                               |

**Source:** Adapted by Schools That Can from *Handbook for Integrating Computational Thinking in Elementary STEM Activities*, Education Development Center, Inc., 2019.
**Maker Pedagogy**

Maker pedagogy emphasizes learning through student-centered inquiry, creating, and innovating. It is based on the principles and practices of the engineering design process—an iterative cycle consisting of defining a problem; researching, planning, prototyping and testing solutions; and refining the solution as necessary (NRC, 2012). In MPP, Maker pedagogy was infused into lessons using a simplified version of the engineering Design Cycle (see Figure 1 below). In each lesson, students created a prototype or a simulation—most often through Scratch¹—to demonstrate their understanding of the scientific phenomena they were studying.

![Figure 1: MPP Design Cycle](image)

In the “Discover Phase” of the Design Cycle, students conducted research on one or several aspects of the scientific phenomena they were studying. Teachers sometimes asked students to write an outline or “storyboard” of their planned prototype or simulation. Then, in the “Create Phase,” students used CS tools to develop that prototype or simulation, drawing on their research and outline/storyboard. Teachers encouraged students to collaborate with one another to solve issues that arose in designing their projects.

In the “Improve Phase” of the Design Cycle, students got feedback from peers and the teacher and used it to iterate on their designs and make their projects better. Feedback took the form of students “playtesting” the prototypes or simulations developed by their peers and often using sticky notes or a teacher-designed template to provide constructive written feedback. Students also may have gotten peer feedback by sharing their projects to a “Scratch studio,” where other students could access and leave comments on the projects. Each phase is designed to foster creativity and innovation, encourage improved understanding of the science and CS/CT knowledge and skills, and develop other essential skills, including persistence, learning from failure, collaboration, and effective communication.

In addition to centering lessons around the Design Cycle, MPP teachers used a variety of other strategies aligned with Maker pedagogy, such as providing ample opportunities for hands-on learning, using Maker language (e.g., referring to students as engineers, discussing the importance of learning from failure, etc.), encouraging peer collaboration, and holding debrief circles (quick check-ins during and at the end of the activity to discuss students’ insights and assist processing and understanding of an idea). (See the Resources section for a description of Maker strategies encouraged by MPP.)

Past research points to the promise of a Maker approach (Vossoughi & Bevan, 2014; Bevan, 2017). Studies suggest that the student-centered, hands-on, iterative nature of Maker activities and pedagogy supports student learning and development by:
• Engaging students in iterative, improvement-focused cycles. In making activities, students develop and test their ideas and learn from their mistakes and the feedback they receive (Blikstein, 2013; Gutiérrez et al., 2014; Petrich et al., 2013; Vossoughi & Bevan, 2014; Vossoughi et al., 2013). The iterative testing cycle, focused on continuous improvement, promotes deeper learning, persistence, and a growth mindset (Ryoo et. al., 2015; Turkle and Papert, 1990).

• Presenting concepts and practices in the context of authentic tasks. Engaging students in creating solutions to actual problems allows for deeper understanding of the material (Blikstein, 2013; Peppler, 2013).

• Fostering meta-cognitive skills that help students consolidate understanding. Having students explain their projects and how they solved problems makes thinking visible and helps solidify learning (Bevan & Ryoo, 2016).

• Cultivating student interest and creativity. Students and teachers find making activities to be fun and exciting, which sparks interest in the content and promotes deeper engagement and learning (Peppler, 2010; Peppler, 2013; NRC 2009; NRC 2012; Ryoo & Kekelis, 2016; Vossoughi & Bevan, 2014).

• Promoting equitable, culturally-relevant learning. Making offers multiple entry points since it does not require significant prior knowledge. It also leverages student interest and cultural resources, thereby appealing to a wide range of students, including those who do not identify as a “science” or “computer science person” (Bevan & Ryoo, 2016; Halverson & Sheridan, 2014; Ryoos et. al, 2015).

Further, Maker activities can address the goals of a variety of science curricula as well as key practices of K-12 science education as articulated in the Next Generation Science Standards (NGSS), such as defining problems, developing and using models to describe phenomena, constructing explanations, and designing solutions. They can also foster core CS practices and 21st century skills, including collaborating around computing, recognizing and defining computational problems, developing and using abstractions, creating computational artifacts, and testing and refining computational artifacts (K-12 CS Framework, 2016). (See the Resources section for a table showing the relationship between CS/CT, NGSS, and Maker dispositions and strategies.)

School and Classroom Context

Sixteen teachers worked with us to integrate CS/CT into their in-school science classrooms. These teachers played a variety of roles at their schools. During the 2019-2020 school year, some were traditional core classroom teachers who provided instruction in multiple subjects (e.g., English, math, science, social studies etc.) while others were technology instructors or science specialists who “pushed into” classrooms. MPP teachers taught students in 3rd through 5th grade. Some implemented this program by themselves; others collaborated with colleagues at the school (see Table 2 below for more information).
Table 2: Characteristics of MPP Teachers

<table>
<thead>
<tr>
<th>School</th>
<th>MPP teachers</th>
<th>Grade Level</th>
<th>Teacher Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Maple*</td>
<td>MS. Hogan</td>
<td>4th grade</td>
<td>Core classroom teacher</td>
</tr>
<tr>
<td></td>
<td>Ms. Massey</td>
<td>3rd grade</td>
<td>Technology/special education teacher</td>
</tr>
<tr>
<td></td>
<td>Ms. Allison</td>
<td>5th grade</td>
<td>Core classroom teacher</td>
</tr>
<tr>
<td>Greendale*</td>
<td>Mr. Drake</td>
<td>4th grade</td>
<td>CS teacher/technology teacher &amp; coordinator</td>
</tr>
<tr>
<td></td>
<td>Ms. Flores</td>
<td>3rd-5th grade</td>
<td>STEM magnet specialist</td>
</tr>
<tr>
<td>Pinewood*</td>
<td>Ms. Huff</td>
<td>4th grade</td>
<td>Science teacher</td>
</tr>
<tr>
<td>Oakwood</td>
<td>Mr. Nieves</td>
<td>5th grade</td>
<td>Science/math classroom teacher</td>
</tr>
<tr>
<td></td>
<td>Ms. Morgan</td>
<td>5th grade</td>
<td>Science/math classroom teacher</td>
</tr>
<tr>
<td>Redwood</td>
<td>Mr. Persaud</td>
<td>5th grade</td>
<td>Core classroom teacher</td>
</tr>
<tr>
<td></td>
<td>Ms. Gonzales</td>
<td>4th grade</td>
<td>Core classroom teacher</td>
</tr>
<tr>
<td>Beechwood</td>
<td>Mr. Chan</td>
<td>5th grade</td>
<td>Core classroom teacher</td>
</tr>
<tr>
<td></td>
<td>Mr. Mehta</td>
<td>5th grade</td>
<td>Science teacher</td>
</tr>
<tr>
<td>Sycamore</td>
<td>Ms. Johnson</td>
<td>5th grade</td>
<td>STEAM magnet resource specialist</td>
</tr>
<tr>
<td></td>
<td>Ms. Walton</td>
<td>5th grade</td>
<td>Science/math classroom teacher</td>
</tr>
<tr>
<td>Hawthorne*</td>
<td>Ms. Cook</td>
<td>3rd grade</td>
<td>Science teacher</td>
</tr>
<tr>
<td></td>
<td>Mr. Conrad</td>
<td>3rd-5th grade</td>
<td>Science teacher</td>
</tr>
</tbody>
</table>

Note: *indicates case study schools.

MPP teachers worked in eight elementary schools, located across four of the five boroughs of NYC. Although all schools served predominantly low-income Black and Latinx students, they varied in size (ranging from 294 to 811 students), the proportion of students who were English language learners (ranging from 3% to 40%), and the percent of students with disabilities (ranging from 10% to 31%). Two of the schools were magnet schools, and two had a special focus such as technology (see Table 3 on the next page). Seven schools used Amplify as their in-school science curriculum. One school used the Full Option Science System (FOSS) program, supplemented with curriculum teachers designed to align with the NGSS.
Table 3: MPP School and Student Characteristics

<table>
<thead>
<tr>
<th>Borough</th>
<th>School 1 Queens</th>
<th>School 2 Queens</th>
<th>School 3 Manhattan</th>
<th>School 4 Queens</th>
<th>School 5 Bronx</th>
<th>School 6 Brooklyn</th>
<th>School 7 Bronx</th>
<th>School 8 Brooklyn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>537</td>
<td>811</td>
<td>441</td>
<td>873</td>
<td>294</td>
<td>406</td>
<td>337</td>
<td>522</td>
</tr>
<tr>
<td>Students with economic need* (%)</td>
<td>77</td>
<td>86</td>
<td>79</td>
<td>87</td>
<td>86</td>
<td>79</td>
<td>59</td>
<td>77</td>
</tr>
<tr>
<td>Asian (%)</td>
<td>14</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Black (%)</td>
<td>10</td>
<td>0</td>
<td>34</td>
<td>1</td>
<td>32</td>
<td>85</td>
<td>62</td>
<td>7</td>
</tr>
<tr>
<td>Latinx (%)</td>
<td>52</td>
<td>98</td>
<td>51</td>
<td>95</td>
<td>65</td>
<td>9</td>
<td>32</td>
<td>84</td>
</tr>
<tr>
<td>White (%)</td>
<td>20</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>English language learners (%)</td>
<td>26</td>
<td>27</td>
<td>5</td>
<td>40</td>
<td>9</td>
<td>10</td>
<td>3</td>
<td>19</td>
</tr>
<tr>
<td>Students with disabilities (%)</td>
<td>22</td>
<td>21</td>
<td>31</td>
<td>23</td>
<td>21</td>
<td>10</td>
<td>29</td>
<td>16</td>
</tr>
<tr>
<td>Special focus of school</td>
<td>None</td>
<td>None</td>
<td>Magnet-Engineering, Architecture &amp; the Arts</td>
<td>Makers and Artists</td>
<td>None</td>
<td>None</td>
<td>None</td>
<td>Magnet STEAM</td>
</tr>
</tbody>
</table>


Note: *Economic Need Index is an estimate of the percentage of students at the school facing economic hardship, based on temporary housing, eligibility for public assistance and Census tract poverty rates.

Regardless of teacher role, grade level, or curriculum, we expected teachers to implement a minimum of three to four CS/CT-integrated lessons per science unit (approximately 9-12 lessons over the course of the 2019-2020 school year). Additionally, we asked teachers to build student assessments of CS/CT skills into each integrated science unit (approximately three to four assessments per year).

How the Partners Worked Together

The three organizations that developed this RPP have different areas of focus and strength. The Research Alliance for NYC Schools, a research center housed at New York University conducts rigorous, mixed-methods studies on topics that matter to the NYC public school system. MakerState develops curriculum, provides professional development, and implements CS and STEM focused programming that is grounded in Maker pedagogy in schools across the City. Schools That Can (STC), a national school support and network organization, promotes learning that is relevant, authentic, hands-on, and connects students to the real world. As shown in Figure 2 below, the two practice partners—MakerState and STC—had primary responsibility for leading recruitment of and communication with schools and teachers, developing curricular materials and resources, and designing and facilitating PD and coaching supports for teachers. The Research Alliance had primary responsibility for collecting, analyzing, and sharing formative and summative feedback with our practice partners, and took the lead in organizing the RPP meetings. All three organizations contributed to identifying problems of practice and possible solutions, developing our research questions and measurement strategy, interpreting data, and disseminating findings and resources from the project.
MPP sought ways to draw on the insights of participating teachers, as well as external experts, in the design and implementation of the programming. The project’s “teacher council” helped adapt the Maker Partnership curriculum and PD supports to an in-school setting. The council, comprised of two MPP teachers and one experienced teacher from outside of the project, used the Understanding By Design (UBD) framework (Wiggins & McTighe, 2011) to refine instructional unit plans and make recommendations for new lessons and units. This group also helped to ensure alignment of the Maker Partnership curriculum with the in-school science curriculum, Next Generation Science Standards (NGSS), and the K-12 CS Framework.

We also engaged an advisory group to inform our work. The interdisciplinary advisory group, with expertise in CS/CT, science, Maker pedagogy, design-based research, teacher education, and RPPs, provided ongoing external, critical review of the project design and activities, including our theoretical framework and plans for data collection, analysis and reporting. The advisory group met four times over the course of the project, and we used their reviews and feedback to assess our progress, improve the project over time, and reflect on the overall success of MPP.

Research Questions, Data, and Methods

The research in our partnership was driven by four overarching questions:

1. How and to what extent did MPP teachers integrate CS/CT into their science instruction?
2. What were the facilitators and barriers to CS/CT integration?
3. What were the outcomes for MPP teachers?
4. What were the outcomes for students in participating classrooms?

We used both qualitative and quantitative data to address these questions, triangulating across data sources to test the consistency of our findings and provide deeper insights about the MPP experience. Our data collection methods included:
Five surveys of teachers (drawing most heavily on our final teacher survey, administered during the summer of 2020)

Four focus groups of teachers

Pre/post surveys of students’ attitudes and beliefs

Pre/post assessment of students’ computational thinking skills

Six interviews with STC and MakerState practice partners

19 observations of instruction and post observation interviews with selected teachers

14 interviews with teachers and administrators in selected schools

The observation data as well as our interviews with teachers and administrators were a part of targeted case studies (Stake, 2004) that we conducted in four schools (Hawthorn, Greendale, Pinewood, and Red Maple). Our aim with these case studies was to explore implementation more deeply and uncover conditions that were promoting or hindering success. We selected the case study schools to represent the range of school and teacher characteristics across the project, including location (Brooklyn, Manhattan, and Queens), size (small, medium, and large), teacher roles (core classroom teachers, CS teachers, and magnet specialist), and focus of the school (magnet and traditional neighborhood elementary schools). All the case study schools served predominantly Black and Latinx and low-income students. In addition, we selected schools that varied in strength of implementation after the first semester. In collaboration with our practice partners, we selected two schools that were very quick to get up to speed and begin implementing MPP, and two that were slower to begin implementing.

To facilitate the use of the data we collected for program improvement, we provided quick-turnaround summaries of the results to our practice partners. We also engaged our partners in collective sense-making of the data, by reviewing summaries and discussing interpretations and implications of the findings. The insights from our collective sense-making led to the lessons learned and recommendations discussed in the final section of this report. See the Appendix for data collection instruments used in this research.

Impact of Remote Instruction on MPP

In March 2020, all New York City schools moved to remote instruction as a result of the COVID-19 pandemic. Most MPP teachers had just begun to integrate CS/CT into their science lessons, and the project had two more PD sessions planned—one in March and one in July. The impact of moving all project and research activities to virtual environments was significant. Below we summarize the impact on the PD and support offered to teachers and on our research. In the Research Findings section, we summarize the impact of the move to remote learning on teachers’ implementation of MPP in their instruction.

Impact on PD and Support

MPP practice partners quickly pivoted to hold the remaining PD sessions remotely over Zoom. We also moved all in-person coaching support to virtual environments (e.g., Zoom, phone calls, emails). Feedback from teachers indicated that the PD was successful despite the move to Zoom. Several teachers noted that it was more convenient for them to attend
the PD virtually because it eliminated a long commute. For example, one teacher said “Logistically, this [meeting over Zoom] is a lot easier than leaving the building for an entire day. It’s more manageable.” The shift also allowed teachers to juggle competing demands. For example, one teacher needed to attend to a request from her principal during the PD. She noted that she appreciated being able to step away for a short period of time to do so. Other teachers noted that learning new skills on Scratch was easier in the remote environment, because screen sharing allowed them to see up close what the facilitator was doing.

Impact on Research

In terms of our research activities, all in-person data collection was discontinued. This included site visits to case study schools to conduct classroom observations and interviews with teachers and school leaders. In lieu of these activities, we were able to observe five asynchronously delivered remote lessons and interview five teachers and one administrator by phone. We also conducted virtual focus groups of teachers following the March and July PD sessions. To collect the student post surveys and assessments, we asked teachers to administer the instruments to their students online. Five teachers were able to do so, though they noted that student response was low. The other teachers did not administer the surveys and assessments, noting that, understandably, it was not a priority given the pandemic.
Research Findings

How and to What Extend Did MPP Teachers Integrate CS/CT into Science?

In this section, we describe how MPP teachers integrated CS/CT into their in-school science instruction, including the frequency and variation in their practices. Throughout this section, we include short vignettes of instructional practice and example lessons based on our case studies to provide illustrations of what integration looked like in practice.

How Teachers Integrated CS/CT into Science

**Frequency.** In our final teacher survey, 9 out of 10 respondents indicated that they had integrated CS/CT into their science instruction before schools transitioned to remote learning due to the pandemic. However, there was a substantial range in how frequently they did so. Of the nine teachers who answered this question, two reported that they integrated CS/CT into one to three science lessons, three teachers reported that they integrated CS/CT into four to six lessons; and two teachers reported that they integrated CS/CT into seven or more lessons. (Two teachers did not respond to this question.) This mirrored what we learned from teacher focus groups: When we asked how often teachers had integrated CS/CT into their science instruction, in January 2020, responses varied from once a month to twice a week. It is also important to note that teachers were not able to begin implementation at the start of the school year because they needed time to create and adapt MPP lessons to align with the science curriculum and context of their specific classrooms. It is likely that if schools had not closed due to COVID-19, teachers would have integrated CS/CT into their instruction to a greater extent by the end of the school year.

**Teachers’ roles.** Given the different roles MPP teachers played in their school, it is not surprising that there was variation in how integration of CS/CT into science was enacted. For example, at one of our case study schools, Red Maple, two of the participating teachers were 4th and 5th grade classroom teachers (Ms. Allison and Ms. Hogan) who taught a common group of students. A science cluster teacher taught students in 4th grade three times per week and those in 5th grade twice per week. Ms. Allison and Ms. Hogan collaborated with the science cluster teacher (who was not part of MPP) to integrate CS/CT into the Amplify science curriculum. They also supplemented the science lessons with additional classroom activities that they planned. In contrast, the third MPP teacher at Red Maple, Ms. Massey, was a 3rd grade special education and technology teacher. She integrated science and CS/CT into all of her 3rd grade classes. All three teachers at this school used Scratch as a CS tool to help students explore and demonstrate their understanding of natural phenomena, such as the water cycle, the solar system, and electrical energy.

In contrast, at our second case study school, Pinewood, the participating teacher was a 4th grade science teacher who taught three classes of students two periods a week. The teacher, Ms. Huff, used the FOSS science curriculum and supplemented it with activities aligned to the NGSS that she created. She also used Scratch as a CS tool to help students explore and demonstrate their understanding of natural phenomena, such as energy, sound, and types of collision.

At our third case study school, Greendale, the participating teachers included a STEM magnet specialist and a 4th grade CS/technology teacher. The STEM magnet specialist
worked primarily with administrators to write grants for school-wide CS programming. The CS teacher, Mr. Drake, saw students one period a week. He coordinated with the 4th grade science and classroom teachers (who were not part of MPP) to integrate CS/CT into the Amplify science curriculum. Similar to other participating teachers, Mr. Drake used Scratch as a CS tool to help students explore and demonstrate their understanding of natural phenomena, such as systems, animal night vision, and Earth’s layers.

The variation illustrated by these three case study schools suggests there is no single “one size fits all” model for who should be responsible for integrating CS/CT into elementary level science (core teachers, science teachers, CS/tech teachers), or where it should occur (in a science class or in a CS/tech class). Rather, integration can occur in a variety of ways and must reflect the specific context, resources, and needs of the school.

**What integrated lessons looked like.** To support teachers, MPP created a model to guide the development and implementation of integrated CS/CT and science lessons (see the model outline in the Resources section). In this model, the first step is to select the science concepts that are to be the focus of integration. This may include coordinating with colleagues (e.g., science, technology, or classroom teachers) to understand where students are in their science learning, and how the integrated lesson can support that learning. Next, teachers prepare a project or series of lessons that are designed to help students use CS/CT to create a simulation of the science concept. For example, for a science unit on food and energy, students might use Scratch to create a program that simulates the growth of food and demonstrate how animals and people obtain their energy from food.

Teachers then infuse a Maker approach into the lesson by structuring it around the Design Cycle (Discover, Create, Improve—see “The MPP Instructional Model” above for a more detailed description). The MPP model also encourages teachers to use other Maker strategies throughout instruction, including think-pair-shares, peer collaboration and feedback, circle meetings, and using Maker terminology, such as referring to students as engineers and encouraging them to take risks and learn from failure. At the end of the activity or project, teachers facilitate a debriefing of the experience, asking students questions such as: What did you learn? What were the greatest successes? What were the greatest challenges? How would you improve your projects with more time? How has the project affected your interest in the science concept, or desire to learn more?

**How Teachers Used Maker Pedagogy to Integrate CS/CT**

Looking across survey, interview and focus group data, we found that most teachers used elements of Maker pedagogy to facilitate the integration of CS/CT into their science instruction. For example, 8 of the 10 teachers surveyed reported using the Design Cycle to frame their instructional arcs. One teacher described how she organized a lesson on energy and motion around the three phases of the Design Cycle:

Students first explored (the “Discover Phase”) the concept of motion with toy cars, rulers, and timers. [They] collected data and asked questions offline. Students then used their knowledge about motion to create a simulation (the “Create Phase”) of [either] what they had done or a simulation of a scenario that they did not exactly try in the Discover Phase. Students shared their simulations (the “Improve Phase”) with each other and the class at large to get feedback.
Another teacher described being very intentional about including the Design Cycle and a Maker approach in her instruction:

> I write it into my [lesson] plan. It’s very deliberate. And I’ll say [to the students], ‘Today we’re going to be doing some discovery around circuits and magnets.’ I’ve tried to really work on the language so that students can say exactly what they’re doing and why.

Similarly, another teacher noted that structuring her lessons around the Design Cycle helped her with planning:

> It reminds me to do a Discover Phase before jumping into the Create Phase. Especially when we’re so short on time, it’s so easy just to jump into, ‘here’s the project, go do it’, without thinking about the reasoning behind it.

Using the Design Cycle helped teachers structure their lessons as authentic tasks—engaging students in creating and testing solutions to problems, and leading to deeper understanding of the concepts and practices being taught (Blickstein, 2013; NASEM, 2021; Peppler, 2013),

In addition to the Design Cycle, teachers drew on other elements of Maker pedagogy in their instruction. The majority (8 out of 10 surveyed) reported that they used circle meetings or debriefs in some of their lessons. Circle meetings and debriefs provide students with an opportunity to reflect on what they learned, solicit feedback from others to iterate and improve on their designs, and present and explain their work to their peers, which fosters meta-cognitive skills (e.g., making thinking visible) that help solidify understanding (Bevan & Ryoo, 2016).

Nine out of 10 teachers also reported emphasizing hands-on learning in most or all of the lessons where they integrated CS/CT into science—as well as getting their students to support or collaborate with one another, using Maker language in their instruction, and using a teacher-built prototype, such as a sample Scratch program, to facilitate student engagement and learning. Additionally, nine teachers reported using student self-assessment rubrics in some of their lessons (see Figure 3 below). Data from our focus groups support these findings—and highlight modifications. For instance, one teacher used reflection (writing) prompts to support or replace circle debriefs.

Not all of the teachers we interviewed found it as easy to employ a Maker approach in their instruction. One teacher from Hawthorn Elementary noted that although they used a lot of hands-on activities, they struggled to cover all three phases of the Design Cycle in their lessons, finding it took more than one period to complete:

> The [lessons] are so time-consuming, that I just let that fall by the wayside. To build that up... I would have had to take more of a concerted effort to explain [the Design Cycle] and go over it in the beginning when we started, and I did not do that.

This teacher went on to note that as a result, their students understood the concept of play-testing, “but they’re not—I don’t think they’ve internalized what those steps of the process are in terms of computer science.”
In the textbox below, we provide a vignette illustrating how one MPP teacher integrated CS/CT into her 4th grade science class. As the vignette describes, Ms. Hogan’s students used Scratch to create simulations of a blackout, demonstrating what they learned about causes and possible solutions in a science unit about electrical energy. Integrating CS/CT into science by having students create a model or simulation showing what they learned was common in the classrooms we observed. Ms. Hogan’s students used several CS/CT practices to develop their programs: They used abstraction and decomposition to create a simple model of complex phenomena (the electrical blackout). They also used abstraction and decomposition when making decisions about which sprites (images that can be created and programmed in Scratch) were needed to create their simulation, what the sprites should do, and where they should go. They used algorithmic thinking to create the code for their program, and iteration to “debug” and improve their simulations by providing each other with feedback and revising their code. Ms. Hogan incorporated Maker strategies by structuring the lesson around the Design Cycle, starting the lesson with a circle meeting, asking parents and peers to provide feedback, and facilitating a circle debrief at the end of the lesson. These activities allowed students to develop, test, and improve on their ideas by learning from their mistakes and incorporating the feedback they received. By having students present and explain their projects to the class, students made their thinking visible, facilitating metacognition. Presenting and explaining their projects was also meant to help solidify their understanding. (See the Resources section for the Ergstown Blackout lesson plan.)
Classroom Observation: Ergstown Blackout

We observed Ms. Hogan's 4th grade class on a day when parents were invited to shadow their children as they engaged in their CS work. Students were tasked with building a model that illustrated causes of and potential solutions to an electrical blackout in Ergstown, a fictional town. The project was part of a science unit on electrical energy.

Using the Design Cycle, Ms. Hogan asked students to work in the “Create Phase” to design a Scratch program that showed why a blackout might occur (e.g., a heat wave that taxes the electrical grid, or a storm that knocks down electrical wires), and ways the town might address the causes of a blackout, for example, through alternative sources of energy. In this phase, students created several sprites and backdrops—images that can be created and programmed in Scratch—and decided where to put them to simulate a blackout. For example, one student created sprites to look like people, and a backdrop that looked like a city skyline. To create the Scratch program, students used a variety of CS/CT practices, including abstraction to create a simplified representation of a blackout, and programming and algorithmic thinking to think through the step-by-step process required to create code for their prototype. Once students had made a prototype of their simulation, they moved on to the “Improve Phase.” Throughout the lesson, Ms. Hogan reminded the students to use their CS vocabulary as they explained their work to their parents (e.g., sequencing, coding, and XY plane). She also provided students and parents with conversation prompts that included questions for parents and students to ask each other (e.g., “How will you add a sprite?”). During this phase, students taught their parents how to “remix” (modify) their simulation by showing them how to alter the program’s code. For example, parents could change the position of the people by altering the XY coordinates of the sprites.

Toward the end of class, Ms. Hogan facilitated a circle debrief, where two students presented their projects to the class, and the class made suggestions for debugging and improving the presenters’ work. One of those students was Emmanuel. When running his Scratch program for the class, he found a bug: All of the sprites began to talk at once rather than in sequence. Ms. Hogan facilitated peer feedback (a Maker practice) by asking the class, “So, what could we debug right here?” Emmanuel stated, “I know one thing I could fix. I will make it so the voices don’t overlap” and then proceeded to show his code to point out where he needed to add in a “wait time” code between speakers. Ms. Hogan also asked students for suggestions on how Emmanuel might improve his program. “What kind of coding could we use to enhance or improve this prototype?” One student suggested, “You could add solar panels to your backdrop.”

As a follow-up to this activity, Ms. Hogan planned to have students write a narrative description of their project.

In the next textbox, we provide another example of how an MPP teacher implemented a CS/CT integrated science lesson into her 5th grade cluster class. As shown in the vignette, Ms. Allison’s students used Scratch to create simulations of a solar system, demonstrating what they learned about the sizes, positions, and colors of the planets. Ms. Allison’s students engaged in abstraction and decomposition when creating their solar system simulation by breaking it down into individual planets, and by breaking down each planet by key characteristics (size, color, and shape). They used algorithmic thinking to identify the steps they needed to take to create the simulation and the code that would translate those steps into a program. By playtesting their program, they identified bugs and iteratively improved their simulations by revising their code. Ms. Allison incorporated Maker strategies by structuring the lesson around the Design Cycle, starting the lesson with a circle meeting, using maker language (e.g., encouraging students to iterate by tinkering with the planets’ color, shapes, and sizes) and facilitating a circle debrief at the end of the lesson.
Ms. Allison also incorporated Maker strategies by asking students to use the “think-pair-share” strategy (turning to a partner to respond to questions). Students who got stuck when creating their Scratch programs were encouraged to ask their peers for help, illustrating one of the collaboration strategies promoted by MPP. This vignette also illustrates how teachers differentiated instruction by offering extension activities for students who completed the task and were ready for more.

Modifying the MPP Lessons

Because there was variation across MPP teachers in terms of the roles they played (e.g., core classroom teacher, push-in CS teacher, etc.), grade levels served, and science curriculum used, adapting the MPP curriculum and resources was an important aspect of implementing the program. Indeed, the PD facilitators were conscious of striking a balance between providing structure and guidance around the lessons (e.g., fully fleshed out lessons that could be replicated in the classroom without adaptation) and flexibility and autonomy to make adaptations as needed. They noted that, on the one hand, many teachers want curricular units and lessons they can implement as is. On the other hand, teachers need the capacity to make necessary adjustments to align the lessons to their particular curriculum, and to the needs of their students. As one of the MPP practice partners explained, “If you have too much guidance, then standards or curriculum change, and then you’ve got to change the guidance…it’s more about capacity building and trying to
create that internal capacity to make the adjustments that need to be made. I’m hopeful that [this approach] is more sustainable.”

In focus groups, MPP teachers described using the MPP lessons as a “jumping off point” for many of the integrated lessons they implemented in the classroom. Noting that developing completely new lessons was too difficult and time consuming, they described “remixing” lessons created by MPP for their students. According to our survey data, 9 out of 10 teachers reported modifying MPP curricular materials and resources to fit the needs of their students to some degree. For example, teachers reported supplementing the lessons with related videos to pique students’ interest, as well as with visuals and physical models. As noted by one teacher:

I added a lot of pictures. When [some students] said they wanted to [design] a bowling alley, I showed them pictures of the inside of a bowling alley, so they get… ideas they can use for a bowling alley when they’re designing it.

As noted earlier, the time constraints of a class period presented challenges—which led to modifications. As one teacher shared, “There is no time to do a full-blown project.” Another addressed this issue by breaking down lessons into smaller units that took less time. A different teacher modified lessons to address the increasing competitiveness she observed among her students by changing independent activities to group projects, thus encouraging productive peer collaboration. Teachers also modified the MPP lessons to differentiate activities for students at different learning levels. For example, teachers frequently scaffolded learning with materials like the question prompts that Ms. Allison and Ms. Hogan used in their classrooms (e.g., What could we do to enhance or improve this prototype?). Another strategy teachers used was strategic student grouping. As one teacher shared:

I had [students] in partnerships based on levels, where I had a more experienced coder with the less experienced one...Another way I [differentiated]—instead of having them complete the project, they might have done a debugging project where the project was made for them already, and they just had to debug the problem. And I would always keep my project up without the code showing so they were able to see how the [project] should actually look without actually seeing the code.

A teacher at Hawthorne also used the strategy of pairing more experienced coders with less experienced coders. This helped her address that fact that she has students at four different grade levels in her classroom—with very different levels of experience using Scratch. Another teacher described differentiating instruction for her low-level readers by reading instructions aloud.

MPP teachers also provided extension activities and encouraged students who finished their projects quickly to embellish their work by adding elements beyond what was required. For example, we observed one classroom in which students were using Scratch to create a simulation of a sunrise and sunset. In this classroom, some students were much further along than others. As a teacher explained in the observation debrief, “Some kids [were] at the point where they’re still working on backdrops, and some kids [were] at the point where they’re ready to finish up and do the extension activity for this.” Students who were further along were encouraged to extend their programs:
If you have already programmed your sun to move and your backdrops are changing, now’s the time to think about some of the ideas you shared before: What kinds of things can you add to show nature? How would I put in a moon or stars? That is your next task.

The modifications and differentiation strategies used by MPP teachers allowed students to work at their own pace and made the classroom activities accessible for students with a wide range of interests, abilities, and needs—all while maintaining fidelity to the key tenets of the integrated lessons as they were designed.

**Implementation during remote instruction.** After schools switched to remote learning as a result of the pandemic, the number of teachers who reported integrating CS/CT into their science lessons dropped substantially: Only 4 of the 10 teachers surveyed reported integrating CS/CT into their remote science instruction. In some cases, teachers reported that their schools did not prioritize science during remote instruction, making integration of CS/CT into science impossible. As one teacher explained:

>I think everything changed with the remote learning, so it was just a totally different world. Anything that was not essential to the basic functioning of the school and just keeping students going… automatically became optional or became something that, “Do it if you can. We’re just trying to survive this experience.”

In other cases, teachers expressed that students’ engagement in CS/CT flagged during remote instruction. Further, teachers struggled to adapt the science curriculum to an online environment. For example, one teacher noted he was not able to find online, student-friendly materials aligned with the Amplify science curriculum his school used. Additionally, teachers struggled with implementing key aspects of the Maker approach virtually, such as the “Discover Phase” of the Design Cycle, where students often worked in pairs or small groups to brainstorm and research their topic. “Student collaboration was stunted, and happened marginally in the comments section [of the online platform] during remote learning,” reported one teacher. Teachers also noted that it was difficult to assess student learning and progress, or even if students were on task, during remote instruction. Finally, teachers reported that implementation of MPP was hindered by the fact that, without physically being together, they could not collaborate as easily with their peers to plan instruction or share physical materials. This was particularly a problem for Mr. Drake, who worked closely with the science teacher to provide the science content. “[The challenge] was teaching them the science component…not being able to do that remotely because I need to learn [the science] myself, and without the help of the science teacher.”

Despite these challenges, teachers said they were able to implement some aspects of Maker pedagogy in the remote environment. For instance, in describing their remote instruction on our final survey, one teacher noted, “I would still refer to [my students] as engineers when creating or building. I would remind them that things do not go right the first time always, and that’s ok. You take a step back and look at it.” Another teacher reported, “I asked students to make things at home with some of the materials I assumed they might have. The best making came in the form of diagrams and video clips.” Additionally, 2 of the 10 teachers who completed our final survey reported that they organized their virtual lessons around the phases of the Design Cycle.
The pandemic and resultant school closures created such unusual circumstances that it is unclear what implementation might have looked like if teachers had not been grappling with such an atypical year. The textbox above provides an illustration of how one teacher used videos to remotely deliver an integrated CS/CT science lesson on types of collisions.

What Were the Facilitators and Barriers to CS/CT Integration?

As part of our research, we explored the facilitators and barriers to developing teachers’ capacity to integrate CS/CT into science instruction. Specifically, we identified components of the MPP PD and supports that were the most useful—and some that were less useful—for building the knowledge and skills necessary to integrate CS/CT into science instruction through a Maker approach. Our data also pointed to several school-based supports that facilitated implementation. In this section, we first describe the MPP supports that teachers found most helpful and other factors that facilitated implementation. We then describe some of the barriers teachers encountered in integrating CS/CT into their science instruction.

MPP Supports

In interviews, focus groups, and surveys, teachers identified in-person PD opportunities, coaching, and the online learning management system developed for this program, STEM CoLab, as the most helpful supports. Each are described below.
In-Person Professional Development. In-person PD sessions were a foundational component of MPP. In total, teachers received eight days of PD, the last two of which were held via Zoom due to the COVID-19 pandemic. All but the last were full-day sessions.

The PD focused on:

- Helping teachers understand CS/CT concepts through the development of their programming skills (in Scratch) and familiarity with technology (e.g., Ozobots);
- Developing teachers’ familiarity with Maker pedagogy and practices (as a part of this process, teachers learned how to use the Design Cycle to facilitate and monitor student progress through the engineering design process); and
- Generating ideas on how to integrate CS/CT into their teaching of science (teachers were provided with resources and time to develop lessons they could use in their classrooms).

(See the Resources section for an overview of the PD content.) PD sessions were also designed to build community among the teachers by encouraging collaboration and providing a space for participants to share ideas and resources. The vignette in the textbox below describes a portion of the April, 2019 in-person PD session for MPP teachers.

MPP In-Person PD, April 2019

During this segment of the PD, the facilitator modeled a typical CS/CT integrated science lesson that used Scratch to simulate the rising and setting of the sun, demonstrating the change in daylight as the earth rotates. This involved using loops to program the movement of the sun, and creating and programming changes in the screen background to show changes in the sky. As the facilitator modeled the lesson, he prompted teachers to envision him as an MPP teacher and to participate as hypothetical students. He also encouraged the teachers to think about how their students would feel engaging in this activity and how they might differentiate or alter instruction for their students.

The facilitator began the model lesson with a circle discussion on the topic. He reminded teachers that the circle discussion aligns with the “Discover Phase” of the Design Cycle. He asked them to think about the colors associated with a sunrise and sunset, as well as the Scratch codes they might use to simulate the movement of the sun.

After these initial questions, the facilitator moved the teachers to the “Create Phase” of the Design Cycle. He prompted them to login to Scratch and to follow along as he programmed the first few lines of code. Teachers then began to make their own Scratch programs. Throughout the lesson the facilitator asked teachers comprehension questions, such as “What block do you think we need next for this code to run properly?” He also asked teachers reflection questions such as, “How would this apply in your classroom?”

Once the programs were created, the facilitator moved teachers into the “Improve Phase,” where teachers reviewed one another’s work and gave feedback (e.g., each teacher offered their partner a few “glows” or compliments/kudos on their program and a few “grows” or suggestions for how to improve).

At the end of the lesson modeling, the facilitator convened teachers into a closing circle to discuss how it felt to participate in the activity, raise clarifying questions on the content and instruction, and brainstorm how they could translate the lesson into their own classroom (e.g., what curricular modifications would be needed, scaffolds for differentiation, etc.). (See the Resources section for a full description of the Sunrise Sunset lesson plan.)

As the vignette above illustrates, PD sessions were hands-on learning experiences that gave teachers opportunities to build knowledge and skills and engage in MPP curricular
activities in the same way that their students would. This allowed teachers to anticipate instructional challenges they might come up against, and adapt lesson plans and classroom activities to meet the needs of their students. As one teacher explained:

> Every time they were showing us something new, there was always... time provided for us to try it. I think that that was super helpful and an eye opener because you don't always get that at PD. I felt like these trainings were very engaging and hands on. Everything that we were being taught and asked to do with students, we were given a chance to try it ourselves, and we were given that time and space to do it. That was really helpful.

Explicit modeling of the lessons also allowed teachers to “see the sequence of a lesson and how it should go.” Furthermore, teachers pointed to the importance of learning in an environment where they had immediate access to support from MPP staff, and time to plan lessons and collaborate with their colleagues. “Planning together is so helpful!” noted one teacher in a focus group. Her colleagues agreed, noting that they would be willing to attend an additional PD session to do more common planning with support from the MPP staff. Another teacher explained:

> It’s a bit [difficult] to get everything in place to leave the building for a day, but once you’re there, I feel like I can give it my full attention. I can go through the activities. I can touch and work with [the materials] and talk to other people, and that seems to get me the most focused.

As this teacher notes, an important aspect of in-person PDs was that they provided a lengthy period of time (a full day), away from school, where teachers could focus on new learning without the distractions and responsibilities of their daily teaching life.

**Coaching.** MPP teachers also benefited from coaching provided by STC and MakerState. The coaches visited each participating school approximately once a month to meet with teachers and administrators and observe instruction. The visits were an opportunity for coaches to provide MPP teachers real-time feedback on their pedagogy and instruction and to troubleshoot challenges. In the second year, the coaches replaced in-person visits with phone and video calls after schools closed in March 2020 due to the pandemic. Below, an STC coach reflects on the benefits of feedback from a supportive outsider:

> When [I visited], then of course that helped surface any lingering questions they [teachers] had, or questions that had built up since the last training. If I could answer them on the spot, I would; if not, it was something I could follow up [on] with them. Sometimes there were questions that didn't have any simple answer, but I was able to be a thought partner, and talk through different options to resolve the issue.... A special advantage of the coach role... is that I was coming in as a supportive outsider... with knowledge of educational pedagogy, the pedagogy and expectations of our program, and also the tools and practices that we were sharing with teachers.... I was seen to be in a very supportive role. I was not seen in an assessment role where there'd be any punitive results from not being perfect. I think teachers were more comfortable to share what was and wasn't working and get feedback.

This was also reflected in debriefs that we conducted with teachers after observing their classrooms. Often, teachers would report that visits from STC coaches had helped them troubleshoot issues from the class we had just observed.
Additionally, visits allowed coaches to obtain critical contextual information about the types of pedagogical strategies and techniques used by teachers, student engagement with MPP classroom activities, and challenges that teachers encountered as they implemented the MPP curriculum. Below, an STC coach explains what he looked for when he visited teacher classrooms:

I was looking for the engagement level of students with the [activities] and... getting a sense of what their computer science and computational thinking abilities were in the classroom. How they were doing the programming.Were they getting comfortable with it? Were they just copying code from a smart board? When questioned, were they understanding what they had written, and then why they’d written it? Were they getting creative? Were they taking initiative to go off on their own? ...Then just looking for things like... Are there hardware or software issues that impact the ability of students to do this?... I was looking for all of those things to get a sense of how well the [classroom] was working for both teachers and students.

Insights gained from these coaching sessions allowed MPP to customize additional MPP supports in ways that were well suited to teachers’ needs. Additionally, whenever possible, the coaches would use the visits as an opportunity to connect with school administrators to provide program updates, share reflections based on their observations, and name any challenges faced by teachers that administrators were positioned to mitigate (e.g., access to laptops or other needed technology).

In-person coaching was supplemented by weekly coaching calls, which provided additional opportunities for teachers to ask questions, receive feedback on their lesson plans, and troubleshoot challenges they faced in the classroom. MPP staff were also available to teachers via email as needed. The in-person visits, regular coaching calls, and email correspondence provided teachers with consistent, ongoing support that helped them stay connected to MPP staff between in-person PD sessions and allowed teachers to receive assistance that was tailored to their particular experiences, classrooms, and students. As one teacher shared, “I communicate more with the folks in the [MPP] program than my administration... I'm never questioning, when is the next PD? Where is the next PD? What are we working on? [The coach] and I have a standing 11:00 phone call every week. That level of communication, I like.”

A chief challenge of these coaching sessions (as well as the in-person PD) was that they required extensive planning and coordination with teachers who have extremely busy schedules. In addition, since teachers were located in schools across the City, travel times to schools ranged from 30 minutes to two hours each way. This meant that many on-site visits required a minimum of five hours to conduct. With monthly visits scheduled for each participating teacher, this component of the MPP programs was enormously time-intensive. Further, teachers themselves had limited time, making it difficult to participate in office hours and conference calls, for example. We discuss teacher time as a barrier to implementation at greater length in the next section of the report.

**STEM CoLab.** MPP used an online learning management system called STEM CoLab as a central hub for instructional videos, example lesson plans, and project prototypes for teachers. The platform also allowed teachers to communicate with each other, a feature intended to facilitate a virtual professional learning community.
Teacher appreciation for the lessons and resources provided on STEM CoLab was widespread. As one teacher shared, “I definitely liked having the materials available there as a jumping off point for a lot of what I was teaching or doing.” Though MPP Staff envisioned STEM CoLab as a dynamic platform that would facilitate teacher collaboration—a space for teachers to share resources and provide feedback on each other’s work—teachers rarely used it in that way. As one MPP staff member reflected:

> The videos on it proved to be very useful, just because that’s the closest thing you can get to an in-person PD... I think STEMCoLab worked as a static resource [as opposed to a dynamic, collaborative platform]....

**Office hours/weekly technical assistance conference calls.** MPP experimented with two other approaches that proved to be impractical for supporting teachers. MPP initially offered virtual office hours—regularly scheduled times when a coach was available by video conference for teachers to drop in to ask questions and get feedback on their lesson plans. Though MPP surveyed teachers to identify optimal times and provided two-hour blocks on different days and times, very few teachers took advantage of them. As one teacher shared, “I haven’t found a good time for the office hours. That’s not something I’ve interacted with. Maybe real time is not a way to do it. Maybe an email back and forth or making it more appointment-based [would be better]...I just know I've never used it, so it's not quite a fit for me.” Similarly, MPP staff had intended to offer regularly scheduled weekly conference calls, an occasion for MPP staff to provide technical assistance focused on CS/CT skills and to address issues and challenges that teachers encountered. However, it proved impossible to find a time when all of the teachers could participate. As a result, MPP discontinued the office hours and conference calls after a few months and focused on email communication, calls scheduled directly with each teacher or team of teachers at a school, and in-person visits to provide support.

**School-Based Supports**

In addition to the PD and supports that STC and MakerState provided, our data revealed two other factors that supported teachers’ ability to bring what they learned into their instructional practices: opportunities for teachers to collaborate and support from school leaders.

**Opportunities for teacher collaboration.** School effectiveness literature consistently points to teacher collaboration as a key facilitator of the implementation and sustainability of reform efforts (Bryk et. al., 2010; McLaughlin & Talbert, 2001). Evidence from our teacher surveys and interviews suggest that this was also the case for MPP. At the end of the program, teachers were asked about the extent to which different individuals supported their work to integrate CS/CT into science instruction. The top response was other MPP teachers at their school, with 8 out of 10 surveyed teachers reporting they were a support to a moderate or large extent. In interviews, teachers expanded on this finding, noting that they benefited from having other MPP teachers in the school with whom they could share resources and co-plan lessons. For example, one teacher said of the opportunity to collaborate at PD sessions:

> It is helpful to have that time, especially with other people who are teaching the same thing. It’s helpful to sit with somebody who is familiar with [the curriculum] and is planning out similar lessons, to be able to do that work.
A colleague added that they appreciated learning from how others in the PD session approached a coding activity:

> I saw the one or two folks at my table, but it’s like seeing a whole class. Every kid would interpret it slightly different, so I might have liked to spend a few more minutes just seeing how everyone else interpreted the [activity] or what they came up with and how they coded it. [That] would be cool. Because I don’t have anyone in my school that I can bounce this off of.

Indeed, the value of support from other (non MPP) teachers in the school building was a common theme. Three of the ten surveyed teachers reported that non MPP teachers at their school had provided moderate levels of support for MPP implementation. Likewise, interviewed teachers from the Red Maple school said they drew support for their work from teachers in their building who were involved in other CS initiatives, notably NYC’s CS4All effort. One of them explained how teachers collaborated across CS initiatives to support each other and work toward a larger school-wide goal:

> We call it the computer science team at our school—we had been meeting about every five to six weeks as a whole team to discuss our planning, discussing what we’re doing with students, discussing technology needs as far as sharing the laptops and everything. We tried to work that out as a team to make sure that everybody had what they needed when they needed it.

The cross-initiative focus was bolstered by the principal’s broader vision for CS/CT instruction at the school. As she explained:

> My ongoing mantra and what [my role is] as a principal… has been making sure that kids see their classroom teachers doing it [CS/CT] and that the classroom teachers have the skillsets to do it, that kids don’t see technology as something separate that they get from a specialty teacher in a separate room like a computer lab…. I think long term, you create a school-wide culture where the expectation for classroom teachers is that we figure out ways to integrate the two, because technology should be integrated into our lives. It’s not something that we should be seeing as a separate skillset…. We’re not there yet, but that’s where I see the school going five years from now.

These findings suggest that opportunities to collaborate, share pedagogical and instructional practices, provide feedback to colleagues, troubleshoot, and share materials and resources supported teachers’ efforts to integrate CS/CT into science instruction.

**Supportive school leadership.** As the quote above illustrates, another factor that seemed to shape teachers’ capacity to integrate CS/CT into science was the amount of support they received from their school leadership. This support took different forms. For example, school leaders promoted teacher collaboration and arranged for teachers to have common planning periods, provided teacher release time to attend PD outside of school, ensured teachers had necessary equipment and internet access, and facilitated teachers fitting CS/CT integration into their instructional schedules. The principal at Red Maple had a strong vision and allocated teachers’ time and resources for school-wide CS. At Pinewood, the principal ensured that teachers had access to technology. As a teacher from Pinewood described:
One thing I love is that [the principal] always talks about getting more technology, getting more laptops, which makes me very happy. He’s very willing [to say], ‘Okay. You’re working on this. How can we help you? What does that look like? How do we get it to more students?’

Similarly, at Greendale, the principal ensured that the CS teacher received access to science-related course materials.

From the MPP practice partners’ perspective, school leader support for the work of CS/CT integration was crucial. This was particularly the case, they noted, for those teachers who did not have colleagues in the building working on similar efforts. Without the support of administrators, or a cadre of other teachers in the building engaged in MPP, they noticed that some teachers found it difficult to stay involved over the two years of the project. In the absences of more widespread support for the work, individual teachers found it difficult to focus on MPP as different initiatives and priorities came up.

Findings from our teacher survey were less definitive about the extent to which a lack of school leader support was a challenge. We found that, across MPP teachers, there was a fair amount of variation in terms of the amount of support that teachers reported receiving from school administrators. Of the 10 teachers who completed the end of program survey, three said their administration supported integration to a moderate extent, and four reported that their administration supported integration to a small extent. The remaining three teachers reported that they received no support from their administration for curricular integration of CS/CT. However, these teachers also reported that a lack of school leader support was not a challenge to their instruction.

**Barriers to Integrating CS/CT into Science**

**Lack of time.** Similar to prior research (Villavicencio, et. al., 2018), the most prevalent barrier to MPP implementation reported by teachers, across various data sources, was a need for more time. In our July 2020 survey, 7 out of 10 surveyed teachers reported that lack of preparation time was an implementation challenge to a moderate or large extent, a sentiment that was echoed in teacher focus groups and interviews conducted in our case study schools. Teachers reported that lack of time became a barrier to implementation in a number of different ways, including limited time to collaborate with other teachers at their school (despite a strong desire to do so, as described earlier), lack of time to plan lessons, and limited available time in their class or lesson schedule for integration of CS/CT. As noted by one practice partner, “Finding the space in their schedule to deliver the integrated lessons was a huge challenge… combined with the challenge to find the additional planning time to really do their own lessons and to adapt it.” For example, elementary students typically have science classes only a few times a week, and those classes might have been the only opportunity for an integrated lesson. Relatedly, other teachers reported that competing priorities made it difficult for them to find time to deliver integrated CS/CT lessons. As noted by a school administrator:

> If, in a given week, the teacher was thinking that she didn’t get through enough math instruction or reading instruction—which, just because of the way we operate in schools, always takes precedence over things—something with CS/CT might have dropped.
Further, many teachers were working with students who had no prior experience with Scratch. This meant that teachers had to provide foundational instruction in Scratch in order to implement integrated CS/CT science lessons, all within a limited window of time. Many teachers noted that it was challenging to get through their school’s required science curriculum without taking additional time to provide instruction in Scratch coding.

Teachers also said the logistics of setting up for an integrated lesson were time intensive (e.g., distributing laptops, logging onto programs, etc.). In our visits to schools, we observed this challenge: At times, the set-up for a lesson could take nearly a fourth of the total class period or more. This was particularly difficult for teachers who pushed into classrooms to provide integrated science lessons. As described by one teacher,

> It’s always time. That was my biggest issue this year. [Students] were always coming [back from lunch] about 5 to 10 minutes into the period, and then we had to wrap up about 5 to 10 minutes before the period was over, cause they have to pack up. My 50-minute period became a half-hour period, and 5 to 10 minutes of that was logging into Scratch, making sure all the computers worked, because the other thing is—with sharing the computers so widespread—sometimes they weren’t charged from the day before or whatever. You’re passing out the computers, and then you’ve got three kids who have a dead computer, and you’ve gotta switch those.

Additional challenges reported by one or two interviewed teachers included the difficulty of creating original lesson plans tailored to their particular curriculum, a lack of support from their administration, administrators’ lack of understanding of the time needed to develop the integrated lessons, limited resources (e.g., laptops), and the lack of opportunities to collaborate with other teachers at their school who were engaging in similar content and pedagogical approaches. One MPP teacher, who was a technology teacher with little background in science, shared in an interview that his lack of science knowledge presented a challenge. As the teacher explained, “The technology end of it has become second nature thanks to MakerState and because I use Scratch now all the time. Teaching [students] Scratch was the easy part. The science was the more difficult end of it.” Lastly, a small number of teachers shared that adapting their lessons to accommodate a wide range of student abilities was challenging. As one surveyed teacher said, “The variety of ability levels with respect to the Scratch program also presented a challenge. Some kids would still be struggling to log on while others have completed the project.” The range in students’ skill level was a challenge that we frequently observed during classroom visits.

Notably, surveyed teachers indicated that the following were not challenges to implementing what they learned through MPP: a lack of classroom management skills, difficulty understanding of how to use hardware or software, and a lack of hardware (see Figure 4 below).
What Were the Outcomes for Teachers?

Improved Understanding of CS and CT

Our survey and focus group data suggests that, by and large, the supports and resources offered by MPP left teachers well prepared to use Maker pedagogy to integrate CS/CT into their science instruction. As a result of the PD and support provided, teachers reported increased knowledge of CS/CT and Maker pedagogy. For example, in a survey conducted before the PD began (in September 2018), six of the 14 responding teachers said they were not at all knowledgeable about CT, and the rest indicated that they had some knowledge of CT. No teachers reported being very knowledgeable about the concept. By the end of the program, all ten of the teachers surveyed reported that their understanding of CT had increased to a moderate or large extent since the beginning of MPP, as had their ability to engage students in CS activities. Relatedly, seven teachers reported that a lack of CS or CT knowledge was not at all a challenge for them as they integrated CS/CT into their science instruction.

Data from our interviews and focus groups largely support the trends we see in the survey data. For instance, one teacher reported, “The things I can do in Scratch … [I] amaze myself some days. I’m like, oh, wait. I know the answer to that. I would not have known two years ago.” Similarly, a PD provider reported that he saw increased ingenuity in teachers’ programming capacity, which he attributed to teachers’ comfort with the material and a sophisticated understanding of Scratch: “In the last six months of the program… I would put up a sample code or sample Scratch simulation, the teachers would all remix it their own way using blocks in their style, which I really appreciate.”
Shifts in Pedagogical Approach: From Instructor to Facilitator

Our data suggest that MPP’s focus on the Design Cycle and other elements of Maker pedagogy helped teachers see themselves as facilitators of learning—where students and teachers co-construct knowledge and develop skills together—rather than instructors whose job it is to transfer knowledge and skills to the student. For instance, in focus groups that we conducted near the end of the second year of the program, one teacher described this shift as: “[A Maker approach] makes me more of a facilitator—instead of “drill and kill” or “chalk and talk”… I teach less basics.” Importantly, teachers reported that they felt that employing Maker techniques made them more effective instructors. This occurred, in part, because of a reframing of their relationship with their students. One teacher noted, “I [now] tell them when I don’t know something. I tell them I’m not the expert and that there is more than one way to do something.” Put differently, teachers were willing to learn alongside and, in some cases, from their students. This focus group data aligns with results from our final survey, in which 7 out of 10 teachers reported that a limited understanding of how to implement Maker learning in science was not at all a challenge for their instruction.

What Were the Outcomes for Students?

Ultimately, MPP was designed to increase students’ CS/CT skills and enhance their learning of subject-area content through the integration of CS/CT into their science classes. In this section, we examine the experiences and outcomes of the students of MPP teachers using multiple data sources, including students’ responses to our pre/post surveys and a CS/CT assessment. In addition, we gathered data on teachers’ perspectives about outcomes for students from interviews, focus groups, and surveys.

As Table 4 below shows, over 330 students received instruction from MPP teachers (and completed a survey) during the 2019-2020 school year. These students were almost evenly split by gender and were largely students of color. In particular, 63 percent of this group identified as Latinx, and 13 percent identified as Black. The students were in 3rd, 4th, and 5th grade, with the majority of students in 4th grade.

As stated earlier, it is important to note that data collection in the second year of our study was severely impacted by the shift to remote learning due to the COVID-19 pandemic. In particular, it was challenging for teachers to administer the MPP end-of-year survey and CS/CT assessment to their students through remote instruction platforms. As researchers and journalists have documented (Barnum & Bryan, 2020; Burke, 2020; Chiu, 2021), student engagement in the first few months of the pandemic was especially challenging as school staff worked to ensure that students had access to the technology needed for remote instruction and as students learned how to interact with their teachers, classmates, and course content in a virtual environment.
<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gender</strong></td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>42.8</td>
</tr>
<tr>
<td>Male</td>
<td>48.8</td>
</tr>
<tr>
<td>Missing</td>
<td>8.4</td>
</tr>
<tr>
<td><strong>Race/Ethnicity</strong></td>
<td></td>
</tr>
<tr>
<td>Asian</td>
<td>6.3</td>
</tr>
<tr>
<td>Black</td>
<td>13.3</td>
</tr>
<tr>
<td>Other</td>
<td>0.6</td>
</tr>
<tr>
<td>Latinx</td>
<td>62.7</td>
</tr>
<tr>
<td>White</td>
<td>8.7</td>
</tr>
<tr>
<td>Missing</td>
<td>8.4</td>
</tr>
<tr>
<td><strong>Grade Level</strong></td>
<td></td>
</tr>
<tr>
<td>Third</td>
<td>19.3</td>
</tr>
<tr>
<td>Fourth</td>
<td>34.3</td>
</tr>
<tr>
<td>Fifth</td>
<td>27.4</td>
</tr>
<tr>
<td>Missing</td>
<td>19.0</td>
</tr>
<tr>
<td><strong>Sample size</strong></td>
<td>332</td>
</tr>
</tbody>
</table>

**Source:** Research Alliance calculations based on data obtained from the NYC Department of Education and MPP student survey.

**Engagement, Collaboration, and Problem Solving**

Data we collected from teachers—whether through interviews, focus groups, or surveys—underscore their enthusiasm for the ways in which their students were engaging with course content, working with one another, and improving their problem-solving skills. In an interview, Mr. Drake from Greendale Elementary described his students’ response to in-person instruction in the following way:

> I would say with the 25 students that I had... three quarters of 'em were super enthused for the [course content]. They really enjoyed doing it. I think they liked the routine. I think they liked the freedom, also, to experiment with different sprites and add different things to their project.

As this quotation illustrates, opportunities afforded by programming in Scratch helped spur student engagement. In fact, many teachers noted that the integration of CS/CT into science enabled students to be creative or artistic while demonstrating their understanding of scientific phenomena. In one focus group, teachers reported that students enjoyed their science classes much more because of the integration of CS/CT into the subject matter.

Our survey data from teachers also indicate that students, for the most part, were engaged with course materials over the course of the year. For instance, 7 out of the 10 teachers who completed the end-of-program survey reported that their students’ engagement in their class had improved over time. Similar proportions of teachers (6 out of 10) reported that students’ knowledge retention improved as a result of science instruction that integrated CS/CT. Likewise, 6 out of 10 reported that their students’ understanding of scientific phenomena improved over the course of the 2019-2020 school year (see Figure 5 below).
The number of teachers reporting moderate or large improvement as a result of the integrated lessons is notable given that, as mentioned earlier, most teachers reported implementing fewer than seven integrated lessons before transitioning to remote instruction, when implementation dropped substantially.

In addition to their engagement with course content, teachers reported that using a Maker approach to integrate CS/CT into science created classroom environments where students were eager to collaborate with one another. This is, perhaps, unsurprising given that 9 out of 10 teachers who completed the July 2020 survey reported that they used elements of peer support or collaboration, a common approach in Maker pedagogy, in most or all lessons where they integrated CS/CT. Our interview data further reveal the richness of these collaborative experiences. As one teacher from Red Maple Elementary reported,

Many times when another student had a question, there would be a kid nearby that would say, “Can I answer? Can I answer?” Or “Can I help them?” as opposed to just that “I take it. I learn it, and I do it myself.” They’re already... wanting to teach and help others with it, which is what we want them to learn. In anything you’re teaching, that’s what we want them to be able to do, is not only do it themselves but teach others how to do it.

Implicit in this teacher’s report is further evidence that, as we described above, teachers in MPP were willing to embrace their role as facilitators and support their students as they helped one another. Data from our focus groups with teachers, too, generally suggested that students were talking to each other more, helping one another, and gaining knowledge or information from one another. In fact, Ms. Hogan described the peer connections in her classroom at Red Maple as “very rich”. And, in the end-of-program survey, 8 of the 10 teachers who responded reported that their students’ communication and collaboration skills had improved to a moderate or large extent over the course of the year.

Finally, teachers reported that their students’ capacity for problem solving increased as a result of CS/CT integration into science. For instance, in the end-of-program survey, 8 of 10 teachers reported that their students’ problem-solving skills had improved to a moderate or large extent. Our focus group data suggest that this might be related to the fact that CS/CT integration was driven by a project-based learning experience. Such an approach enabled students to take ownership of their work and create projects that authentically represented their skills and interests. Data from our interviews suggest that students’ abilities to take risks also contributed enhanced problem-solving skills. As Ms. Huff at Pinewood Elementary described: “I really like... how willing [the students] are to take risks, and try stuff. Like, ‘Oh, if I add this code, what happens?’ It’s cool if it doesn’t work, to them. Sometimes, it’s even cooler if it doesn’t work the way they initially planned.” The ability to find a variety of solutions to problems enables students to proceed without feeling intimidated and gives them confidence that they will find a solution. Ms. Huff noted that her students were generally able to explain why something did not work—and what they might do differently in the future: “Incorporating CS/CT [into science] helps students to think more deeply and see what they are doing.”
CS/CT Skills and Dispositions

The most straightforward approach to measuring the ways in which integrating CS/CT into science shaped students’ outcomes is to assess differences between students’ CS/CT skills at the beginning and end of the year. As we described earlier, the shift to remote learning on account of the COVID-19 pandemic profoundly impacted our data collection efforts—but we do have data from the beginning and end of the 2019-2020 school year for 23 students.

We created our assessment of CS/CT skills by drawing on and modifying items from three existing assessments: Wiebe and colleagues’ (2019) computational thinking assessment, items available through the Bebras Computing Challenge, and items from Grover’s (2020) assessment, Variables, Expressions, Looping, and Abstraction (known as VELA). Our assessment consisted of nine multiple choice items and one open-ended question. It was administered online to students across the schools in our study. Table 5 below shows the concepts covered in each item and the proportion of students who correctly answered each question at the beginning and end of the year.

In this group of students, we find that end-of-year scores were higher than beginning of year scores: Students correctly answered about a third of questions on the pre-assessment; their responses to just over 40 percent of questions on the post-assessment were right. These improvements were driven by students’ correct responses to items measuring students’ debugging (Q1) skills, their understanding of loops (Q2, Q3, and Q7), and, to some extent, conditional statements (Q4 and Q7).
### Table 5: Percent Correct on MPP Assessment, by Question

<table>
<thead>
<tr>
<th>Question</th>
<th>Pre-Assessment</th>
<th>Post-Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1: Algorithmic thinking, debugging</td>
<td>39.1</td>
<td>60.9</td>
</tr>
<tr>
<td>N=23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2: Algorithmic thinking, loops</td>
<td>78.3</td>
<td>91.3</td>
</tr>
<tr>
<td>N=23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3: Algorithmic thinking, loops</td>
<td>30.4</td>
<td>39.1</td>
</tr>
<tr>
<td>N=23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q4: Algorithmic thinking, conditional statements, loops</td>
<td>22.7</td>
<td>31.8</td>
</tr>
<tr>
<td>loops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q5: Algorithmic thinking, sequencing</td>
<td>13.6</td>
<td>22.7</td>
</tr>
<tr>
<td>N=22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q6: Algorithmic thinking</td>
<td>61.9</td>
<td>57.1</td>
</tr>
<tr>
<td>N=21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q7: Algorithmic thinking, conditional statements, loops</td>
<td>23.8</td>
<td>42.9</td>
</tr>
<tr>
<td>loops</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N=21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q8: Algorithmic thinking, variables, conditional statements</td>
<td>40.0</td>
<td>35.0</td>
</tr>
<tr>
<td>N=20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q9: Algorithmic thinking, abstraction</td>
<td>15.0</td>
<td>10.0</td>
</tr>
<tr>
<td>N=20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All multiple choice questions</td>
<td>36.1</td>
<td>42.8</td>
</tr>
<tr>
<td>N=20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Source:** MPP student assessment administered in Fall 2019 and Spring 2020.

**Note:** We report the number of responses by item. The assessment also contained an open-ended item, whose results are not included in this table. We did not find statistically significant differences between the close-ended pre- and post-assessment results. This may have been due in part to the small sample size.

There are some crucial caveats to consider when interpreting these assessment results. Not only is this a small subset of students who participated in our study, the students who took the pre and post assessments may not be representative of all students in the study. These students may have been more engaged in the material in general, and therefore more motivated to complete the end-of-year assessment remotely. Further, they also had to have the technological capacity and fluency to do so without the aid of their teachers.

That being said, there is other evidence that points to an improvement in students’ CS/CT skills. For instance, focus group and interview data suggest that students’ programming and CT skills improved over the course of the year. A teacher at Greendale reported: “They definitely have algorithms down, and many of them understand the concept of loops too. [And] conditional statements. They understand that too.” Another teacher from Red Maple noted how she was seeing students transfer computational thinking to other subjects: “I started seeing a difference in [students’] reasoning…That is big. Then [I started seeing students’] computational thinking in ELA and math.” Further, in the survey administered in
July 2020, 7 of the 10 teachers who responded noted that their students’ confidence in coding as well as their confidence in CT increased over the course of the year.

Although students’ CS/CT skills are important, so, too, are their dispositions and attitudes toward learning science, CS, and using a Maker approach. To capture these elements—and, importantly, to see if there were any shifts in students’ dispositions and attitudes over time—we administered a survey to students at the beginning and end of the 2019-2020 school year. As with our assessment data, we have a small group of students (N=35) for whom we have survey data from both points in time. For this group of students, we do not see any meaningful shifts in their CS or science confidence or their interest in CS. Although fewer students (74 percent relative to 94 percent) reported feeling excited about science class at the end of the year, about 77 percent agreed or strongly agreed that participating in their science class during the 2019-2020 school year made them excited about learning CS. It’s important to note that the decrease in excitement about science class may be related to a loss in enthusiasm for school more broadly because of the move to remote instruction in the spring of 2020.

Survey items capturing students’ dispositions toward Maker-type activities also yielded mixed results. For instance, at the end of the year, fewer students reported that they liked to take apart or build things to see how they worked (74 percent relative to 91 percent), but students did appear to be more open to feedback. In particular, the proportion of students reporting that they did not like it when others suggest changes to their work decreased from almost half to 20 percent. Furthermore, there did seem to be some movement in students’ growth mindsets—while 86 percent of students agreed or strongly agreed, at the beginning of the year, that excellence in CS skills was static, only half of all students agreed with that sentiment by the end of the year. Despite this improvement, troubling trends remain, especially in our students’ responses to items about gender equity in CS. For example, the proportion of students agreeing with the statement, “Girls can do just as well as boys in computer programming” decreased from 94 percent at the beginning of the year to 71 percent by the end of the year.

Finally, it is important to note that the survey results are subject to the same caveats as the assessment results. The sample size is small and may not be representative of all students in the study, especially in terms of engagement and access to technology.
Lessons Learned and Recommendations

Drawing on multiple data sources, our study produced lessons about building teachers’ capacity to integrate CS/CT into science through Maker pedagogy, about the facilitators and challenges to integrating CS/CT into science instruction, and about students’ experiences in such learning environments. These insights and the recommendations we highlight below emerged from the collective sense-making around the research findings that we engaged in with our practice partners.

What We Learned About Building Teachers’ Capacity to Integrate CS/CT into Science

Developing the Capacity of Novice Teachers to Integrate CS/CT

Integration of CS/CT into science can take many forms and looks different depending on the goals of the instruction, emphasis on content to be taught and level of integration (e.g., more CS/CT or more science), and available tools and resources. Most MPP teachers integrated lessons by using Scratch to create a simulation of a natural phenomenon (such as the water cycle, energy, or the movement of the earth and sun). Teachers may need more PD and support to integrate CS/CT into science using more sophisticated skills and tools, for example by conducting physical experiments, modeling complex phenomena, and using technology to organize and display data. Relatedly, the cost of some hardware and software (e.g., Ozobots) may prevent teachers from expanding beyond Scratch, which is free and easily accessible.

Nonetheless, our experience with MPP demonstrated that teachers with little or no prior experience were able to learn CS/CT concepts and basic to intermediate Scratch programming, and feel comfortable implementing what they learned in their instructional practices within a fairly short period. Notably, we found that teachers needed some facility, but not advanced Scratch skills, to use it as a tool for integrating CS/CT into science. Basic proficiency supported students’ development of basic skills, allowing them to use Scratch to simulate natural phenomena. This bodes well for efforts that are attempting to bring CS/CT integration to elementary students at large scale, especially since many teachers with no prior CS training or credentials are being asked to teach CS (NASEM, 2021).

Types of PD Supports That Were Most Helpful

In testing and iterating on various types of PD and supports, we found that teachers benefitted from PD that met them where they were in terms of their needs, knowledge of CS/CT, and skills, and progressed over time as these needs and skills evolved. They developed skills and comfort with CS/CT and Maker pedagogy at varying rates and required different levels and types of support to continue building capacity. Relatedly, teachers benefited from sustained engagement in PD with sessions spread throughout the year. This allowed them to learn new concepts, approaches, and skills in a session, try them out with their students, and then come together again to review their successes and challenges and get more feedback and support. It also allowed teachers to provide formative feedback to our practice partners, which helped improve the PD and support being provided.
Teachers found a number of characteristics of the in-person PD to be particularly effective. These included **hands-on learning** (e.g., teachers learned CS/CT concepts and skills by creating Scratch programs), **modeling of lessons and pedagogical approaches**, **experiencing the lessons as students would**, **guided practice in using new CS/CT skills**, and **time for planning and collaboration** with other teachers. In addition, we found that the face-to-face training fostered a community of practice and collaboration among teachers that provided mutual support, sharing of resources and plans, and lessons learned. Due to COVID-19, our last two PD sessions were held remotely over Zoom. Because the teachers and our practice partners had established a relationship and strong rapport with each other, the PD was not hindered by the virtual format. Conversely, some teachers reported appreciating the remote format. It eliminated lengthy travel time to the training site, allowed for better visualization of Scratch programming demonstrations through shared screen technology, and allowed teachers some flexibility to attend to time-sensitive issues that arose during the training (e.g., when a principal needed information).

MPP coaches provided **“wraparound” supports** (such as site visits to conduct classroom observations, provide feedback, and meet with teachers and school leaders; regular email communication; and conference calls with each school), which supplemented the knowledge and skills teachers learned during PD sessions. We found that these activities enhanced coaches’ understanding of the contexts in which teachers were working—including a sense of their school’s technology infrastructure, colleagues’ and administrators’ support for each teacher’s effort to integrate CS and CT into science, as well as the school’s overall CS culture and ethos—making it possible to provide better, more customized support.

We also learned that teachers preferred support that addressed their specific needs as they arose (e.g., via email or phone calls), rarely taking advantage of regular opportunities to participate in scheduled sessions (such as office hours) for support. Providing individualized support requires more effort, however, which has time and cost implications that may be a barrier to scaling up.

**What We Learned About Integrating CS/CT into Science Through Maker Pedagogy**

**Factors that Affected Implementation**

MPP teachers played a variety of roles in their school. Some were classroom teachers who taught core subjects to a single group of students. Others were science or technology specialists who either pushed into the classroom, or taught their subject during specific periods. Integration of CS/CT into science instruction looked different depending on these context factors. Schools use a variety of different models to deliver science instruction. There is no “one size fits all” model for who should be responsible for integrating CS/CT into elementary level science (core teachers, science teachers, CS/tech teachers), or where it should occur (in a science class or in a CS/tech class). Rather, integration can occur in a variety of ways and must reflect the specific context, resources, and needs of the school.

**Providing teachers with resources and materials** (e.g., lesson plans and units, model projects and skill-building videos for students, student assessment rubrics, unplugged activities, coding cards) saved them time and effort, and avoided the need to start from
zero. Making these resource available increased teachers’ ability to integrate CS/CT into science instruction. However, even with access to high-quality resources, teachers needed to modify or adapt most activities and lessons to differentiate supports and address the specific needs of their students, as well as to address specific science curricular units they planned to teach. Teachers benefitted from structured time for this work during MPP PD sessions—as well as the guidance and support of MPP coaches and teacher colleagues.

Beyond issues related to implementing the curriculum and pedagogical approach, we also learned that significant logistical issues related to schools’ technology infrastructures (e.g., hardware and internet access, setting up Scratch accounts and getting students logged in, etc.) had to be addressed for successful integration of CS/CT into science. Teachers reported that a top challenge to implementation was time, including the time it took to get the necessary materials set up and to work through the logistics of logging on—which took precious time away from instruction. Teachers benefitted from specific support and assistance troubleshooting and establishing routines and procedures to mitigate these issues.

Another factor that facilitated teachers’ learning and implementation of new content and pedagogy was collaboration with peers. Teachers particularly valued opportunities to work together to design lessons, share pedagogical and instructional practices, provide and receive feedback from colleagues, troubleshoot, and share materials and resources. Working in a school that has an overall vision for CS/CT and concurrent CS/CT efforts (e.g., the districtwide CS4All initiative) seemed to support implementation of MPP by creating an environment where teachers have a community of peers with whom they can share resources and collaborate. These findings align with prior research showing the benefits of collaboration and support for teachers engaged in school improvement efforts (see for example, Bryk et al., 2020; Cochran-Smith & Lytle, 1999; Jaffe-Walter & Fancsali, 2020; McLaughlin & Talbert, 2001).

Non-science teachers in particular (e.g., CS or technology cluster teachers) benefitted from collaborating with science teachers to support integration. MPP PD and supports focused on building skills and knowledge about CS/CT, Maker pedagogy, and integration across disciplines. It presumed teachers had science content expertise and experience teaching science. However, this was not always the case. One of the teachers in our project was a CS teacher with no background in science. That teacher was able to address this gap by working closely with colleagues who could provide guidance on the science content.

Finally, engaging school leaders in MPP was challenging and time consuming, but proved to be an essential element in successful implementation and sustainability. School leaders are in a position to support teachers in overcoming common barriers, for instance, by facilitating class scheduling to allow for integration of CS/CT, carving out planning time with other teachers integrating CS/CT, allowing teachers release time so that they can attend PD, and ensuring that teachers have access to adequate internet, hardware, software, and other needed materials.

The Value-Add of a Maker Approach

Making and Makerspaces have a long history in education, but more recently have been seen as a promising approach to integrating CS/CT into subject areas and broadening participation in CS (NASEM, 2021). In this project, we found that using a Maker approach
provided teachers with a common language and consistent framework for planning and delivering instruction that facilitated student engagement, creativity, collaboration, persistence, reflection, and independence. This framework helped teachers’ shift their pedagogical approach from instructors transferring knowledge and skill to students, to facilitators of learning. Structuring lessons around the Design Cycle allowed students to practice the steps that scientists take in exploring natural phenomena (e.g., gathering information and brainstorming solutions, creating prototypes or models of the solution, testing out and improving the design). Finally, teachers found Maker pedagogy to be an effective pedagogical approach for integrating CS/CT into science because it allows multiple entry points for students with a wide range of CS/CT skill levels.

**Student Experiences**

**Building CS/CT competencies.** Integrating CS/CT into science requires some fundamental CS skills for students. For example, students needed foundational Scratch skills (how to create backdrops and change sprite costumes, how to create simple programs using motion, control, and event blocks) to be able to complete MPP integrated lessons. Similar to teachers, students were able to fairly quickly learn block-based programming, such as Scratch and OzoBlockly, through scaffolded and guided instruction. Peer collaboration (a key Maker pedagogy practice) seemed to facilitate students’ rapid uptake of coding skills and helped teachers address a wide range of CS abilities in their classes.

Nonetheless, students in our teachers’ classrooms spanned the spectrum in terms of CS/CT skills, and many had little or no prior experience. This raises important questions about whether elementary level science teachers should collaborate with computer science or technology teachers to provide CS/CT instruction to students, and whether CS/CT skills should be a prerequisite to an integrated class. If students do not already possess foundational CS/CT skills, they either need support outside of class (e.g., through a separate CS class) or teachers need additional classroom time to build those foundational skills as a prerequisite to integrated CS/CT activities. The variability among our MPP teachers in terms of their teaching responsibilities (e.g., as a core classroom teacher, as a cluster science, or CS teacher who pushes into classes, etc.) showed us that there are several models for how CS/CT can be integrated into science instruction. Depending on what type of teacher is responsible for delivering integrated instruction, students may need additional support to learn the foundational programming skills that will allow them to use CS/CT to explore scientific phenomena.

**Outcomes for students.** Teachers observed that MPP activities were engaging for students, encouraged and offered multiple opportunities for collaboration and providing peer feedback, and improved problem-solving skills—suggesting promising outcomes for students. Our pre/post assessment of student learning and attitudes showed mixed results, and should be interpreted with caution due to low response rates and complications and constraints related to the abrupt transition to remote learning resulting from COVID-19. Additional research, including direct measures of student learning and attitudes, is needed to understand if and how CS/CT integrated into science instruction through a Maker approach improves student learning in either content area. Further, it will be important to study how much exposure to integrated lessons is needed to influence student outcomes.
Recommendations

Emerging from our research and the experience and insight of the MPP partners, we offer several recommendations for PD providers, school leaders, and teachers interested in integrating CS/CT into science using a Maker approach.

For PD providers:

- Collect quick turn-around data on teachers’ development of CS/CT skills and pedagogical practices so that PD sessions can better meet their needs.
- Supplement in-person PD with support between PD sessions (e.g., phone calls, emails) to address teachers’ specific needs and questions as they attempt to implement what they learned.
- Provide teachers with high-quality lesson plans and resources that are aligned with the school’s science curriculum (and that they can easily modify), especially when teachers are learning a new content area, such as CS, with little or no prior experience.
- Prepare teachers to address logistical challenges, such as getting students set up on equipment, logging into computers and Scratch, and storing and maintaining the equipment.
- To the extent possible, work with school leaders and teachers to schedule all PD sessions for the year before the school year starts. This helps leaders and teachers avoid conflicts and preserve the time needed to attend. In addition, consider a mix of shorter, more frequent sessions, and online sessions that may be less burdensome for teachers to attend.
- Provide school leaders with guidance about how to support teachers’ development of integration and Maker practices. For example, coaches could conduct observations with school leaders and provide a rubric they can use to offer feedback to teachers.

For School Leaders:

- Facilitate scheduling, common planning time, and collaboration with other teachers, and address the infrastructure needs that are common barriers to integrating CS/CT into science. Teachers need these resources before and during the school year.
- Integration is inherently multidisciplinary. Therefore, including both science and CS/technology teachers from the start in a collaborative effort will help make the best use of staff knowledge and resources, and create a foundation for scaling within the school.
- Given the benefits of collaboration among teachers, such as providing mutual support and sharing of resources, plans, and lessons learned, consider implementing efforts such as MPP with multiple teachers. Align and collaborate with other CS-related efforts in the building to foster a community of practice.
- Provide teachers with ample time to revise and adapt lessons and projects to address their specific contexts (e.g., to differentiate supports for students, accommodate special needs, and meet students where they are in terms of prior CS experience, and to align lessons with the school/district science curriculum).
• Provide teachers with opportunities to try out new pedagogical approaches and lessons in a lower-stakes environment (e.g., absent of high-stakes teacher evaluations).

For Teachers:
• Use existing MPP lessons and lesson templates (see Resources section in this report) as a launching pad for adapting and developing your own integrated lessons that take into consideration the needs, skills, and interests of your students and alignment to the classroom science curriculum.
• Anticipate and try to address logistical issues (such as having equipment in the classroom, charged and ready to use; providing students laminated login instruction cards to assist with logging into computers and Scratch; testing internet connections and software). Avoid cutting into instructional time by tackling these issues before students arrive for class.
• Work with school leaders to schedule common planning periods or times to collaborate with other teachers to share support, information, and resources.
• Build in “unstructured” coding time to allow students who are ready to go farther in their explorations to do so, and to allow students to express their creativity. This also provides opportunities for teachers to work individually with students who need extra support.
• Use peer collaboration so that students can help each other learn (particularly programming skills). Peer collaboration helps students iterate and improve their own work and develop independence from the teacher, and allows the teacher more time to work one-on-one with students who can benefit from additional support.

Next Steps
MPP was designed as a long-term collaboration that “leverage[s] research to address persistent problems of practice” (Henrick, et. al., 2017, pg. 1). As such, we are aiming to continue our partnership and build on the work we did through MPP to tackle new problems of practice. Thus far, our RPP has focused on building the capacity of individual teachers to integrate CS/CT into science instruction. However, we recognize that, to successfully reach all elementary students, CS/CT must be integrated into core subjects on a schoolwide basis. Doing so increases access and reduces disparities in participation caused by students opting in or out of CS. Implementing throughout a school also avoids the common problem of initiatives fading out as individual teachers change grade levels, subject areas, or schools. Building on the lessons from MPP presented in this report, we hope to design and test a schoolwide model for CS/CT integration. This work will address the pressing need for models that effectively engage underrepresented students in CS and will continue to inform efforts at CS/CT integration more broadly.
Resources

The following resources are drawn from the materials developed and adapted by Schools That Can and MakerState for this project. These resources may be useful to teachers, school leaders, and PD providers engaged in similar efforts.

MPP Professional Development: Overview of Content

The table below describes the Maker strategies, CS/CT topics, and science content covered by our practice partners in each professional development session for Year 2.

<table>
<thead>
<tr>
<th>Professional development dates and link to agenda</th>
<th>Topics Covered</th>
<th>Science Content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maker Strategies</td>
<td>CS/CT Topics</td>
</tr>
<tr>
<td>PD # 1</td>
<td>Review MPP Lesson Plan Template and how it supports student learning</td>
<td>• Review of CS/CT integration in science (2nd Unit) • CS Skill Progressions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PD # 2</td>
<td>Demo Lesson: Experience a CS/CT integration lesson from a student’s perspective: • Discover / Create / Improve Phases • Maker pedagogy • Review how the Design Cycle and maker pedagogy can support students</td>
<td>• CT: abstraction, algorithmic thinking. • Practice Scratch programming skills.</td>
</tr>
</tbody>
</table>
| PD # 3 (Virtual) | Demo Lesson: Experience a CS/CT integration lesson from a student’s perspective:  
- Discover / Create / Improve Phases  
- Maker pedagogy  
- Review how the Design Cycle and maker pedagogy can support students | • CT: Use abstraction, algorithmic thinking, to create models that can be used to simulate how natural phenomena occur  
• CS: Basic block-based coding skills, loops, conditionals, and variables.  
• CT: Lesson Plan on using variables and data  
• CT: Teacher CT Assessment | • Grade 3: Environments and Survival  
• Grade 4: Earth’s Features  
• Grade 5: The Earth System |
| PD # 4 (Virtual) | Review of Model of CS/CT integrated lesson  
Review of how the MPP Design Cycle can support CS/CT integration into science lessons  
CS skills: basic block-based coding skills, loops, conditionals, and variables.  
CT practices: abstraction, algorithmic thinking skills used when creating models that can be used to simulate how natural phenomena occur. | Teachers reflect on the past year’s implementation of CS/CT integrated science lessons. | |

**General Model of Integrated CS/CT and Science Lessons**

*This document* describes the general MPP model for developing integrated CS/CT lessons that use a Maker approach.
## Description of Maker Instructional Models

The table below describes the key Maker instructional strategies that teachers were encouraged to incorporate into their lessons.

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle meeting</td>
<td>The session starts with a class circle meeting where teachers 1) check-in with students/have a warm-up activity, and 2) introduce the topic of study or assignment.</td>
</tr>
<tr>
<td>Design Cycle</td>
<td>The Design Cycle is a continuous cycle of researching, designing, creating, and improving. In the <strong>Discover Phase</strong>, students ask questions to gather information, imagine possible solutions (brainstorm), and plan out their solution. In the <strong>Create Phase</strong>, students take their best solution idea and create a prototype or model, either in physical form or on the screen. In the <strong>Improve Phase</strong>, students test out, debug problems, iterate and improve their design. The cycle goes on as long as necessary, and each time something new is learned, regardless of failure or success.</td>
</tr>
<tr>
<td>Collaboration</td>
<td>Facilitate collaboration by: 1) employing a think-pair-share or small group discussion to highlight/reinforce learning, 2) encouraging collaborative brainstorming, and 3) reinforcing positive collaborative behavior and highlighting the positive outcomes of collaboration.</td>
</tr>
<tr>
<td>Peer feedback</td>
<td>Use protocols to foster peer-feedback (e.g., “<strong>TAG</strong>” —Tell something that you liked, Ask a question, Give a suggestion).</td>
</tr>
<tr>
<td>Maker language</td>
<td>Use the language of Maker learning when engaging in instruction, e.g., refer to students as engineers or Makers; refer to phases of the Design Cycle; use terms and phrases such as prototypes, collaboration, student risk taking, improving on design/iterating, persisting through challenges, and learning from failure.</td>
</tr>
<tr>
<td>Circle debriefs</td>
<td>Quick check-ins during and at the end of the activity to discuss students’ insights and support processing and understanding of an idea. E.g., ask, “What was your best mistake? What was surprising? What was challenging? How did you improve?”</td>
</tr>
</tbody>
</table>
## Alignment Between CS/CT Concepts, NGSS standards, and Maker Dispositions and Attitudes

<table>
<thead>
<tr>
<th>CS/CT Concepts</th>
<th>NGSS Standards</th>
<th>Maker Dispositions and Strategies that are Dimensions of CT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstraction</td>
<td>• Asking questions (science) and defining problems (engineering)</td>
<td>• Confidence in dealing with complexity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Persistence in working with difficult problems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Tolerance for ambiguity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The ability to deal with open ended problems</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• The ability to communicate and work with others to achieve a common goal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Designing for the end user: Empathizing with the problem/challenge of an end user, brainstorming</td>
</tr>
<tr>
<td></td>
<td></td>
<td>solutions and testing those solutions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Debugging: Fixing what doesn’t work</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Testing prototypes: Testing and gathering data on prototype usefulness/effectiveness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Iterative design: Revising, improving and innovating on designs and prototypes</td>
</tr>
<tr>
<td>Modeling and Simulation</td>
<td>• Developing and using models</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Planning and carrying out investigations</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>• Analyzing and interpreting data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Using mathematics and computational thinking</td>
<td></td>
</tr>
<tr>
<td>Algorithms &amp; Programming</td>
<td>• Developing and using models</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Using mathematics and computational thinking</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Constructing explanations (science) and designing solutions (engineering)</td>
<td></td>
</tr>
</tbody>
</table>

**Sources:** Adapted by Schools That Can from The International Society for Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA), 2011; Next Generation Science Standards, 2011; and Handbook for Integrating Computational Thinking in Elementary STEM Activities, Education Development Center, Inc., 2019.
### Scratch Skills and Progression Table

<table>
<thead>
<tr>
<th></th>
<th>Integration 1</th>
<th>Integration 2</th>
<th>Integration 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit 1</td>
<td>Use</td>
<td>Modify</td>
<td>Create</td>
</tr>
<tr>
<td>Unit 2</td>
<td>Loops</td>
<td>Conditionals</td>
<td>Programming challenge (assessment)</td>
</tr>
<tr>
<td>Unit 3</td>
<td>Variables: Creating and setting variables</td>
<td>Variables: Using variables in your programs</td>
<td>Programming challenge (assessment)</td>
</tr>
<tr>
<td>Unit 4</td>
<td>Culminating project/simulation</td>
<td>Culminating project/simulation</td>
<td>Culminating project/simulation</td>
</tr>
</tbody>
</table>

### Lesson Plans

The exemplary lessons below were created by Schools That Can and MakerState:

- [Ergstown blackout lesson plan](#)
- [Sunrise, sunset lesson plan](#)
- [Water cycle lesson plan](#)
  - [Scratch program prototype](#)
- [Variations in a species lesson plan](#)
- [Solar power simulation lesson plan](#)

### Student Rubric

The [Scratch Mastery Rubric](#) is an example of a formative assessment tool created by Schools That Can and MakerState for MPP.
Endnotes

1 Teachers most likely used Scratch as the CS tool to create prototypes because it is a widely used and easily accessible free software.

2 We set these expectations prior to school closures and the move to remote instruction due to COVID-19. Given the challenges related to COVID-19, we did not hold teachers to these expectations.

3 All school, teacher, and student names in this report are pseudonyms.

References


The Research Alliance conducts rigorous studies on topics that matter to the City’s public schools. We strive to advance equity and excellence in education by providing nonpartisan evidence about policies and practices that promote students’ development and academic success.