Over the past decade, the question of how and why to integrate computers into our education has become increasingly prominent.

While much of the conversation has focused on the insufficient STEM career pipeline, the benefits of bringing computers into our schools extend beyond creating the next generation of computer programmers. Early research suggests that computational thinking and computing education have the potential to develop students’ higher-order thinking, problem-solving, collaboration and communication skills in ways that can advance learning across the curriculum and empower students to be creative inventors with technology. Initial studies also point to the connection between computing education and greater student confidence in and engagement with core academic content, including among students struggling academically.

To date, however, the focus has been on building these computing and computational thinking skills at later ages and out of school. That orientation may lead us to miss critical opportunities. Emergent studies suggest that early computational thinking experiences could be the key to developing strong foundational thinking and computing skills, which can then lead to deeper learning more broadly. These experiences also contribute to positive attitudes toward computing, which are critical to keeping students, especially girls and youth of color, open to ongoing experimentation with computers.

Integrating computational thinking and computing education across the curriculum is not only designed to provide students with more time to learn these crucial skills; this approach may make it possible for students to engage with core content in new and creative ways. From an equity perspective, integration also holds the promise of ensuring all youth have access to computing education, not just those with access to enrichment opportunities.
Computing education is receiving tremendous attention, both nationally and in New York. In 2015, New York City launched its Computer Science for All (CS4All) effort with the goal of providing every student with meaningful, high-quality computer science education at each school level—elementary, middle, and high—by 2025. Building on the momentum of that effort, in March 2018 New York adopted a Computer Science Teacher Certification and established a Computer Science task force. Although the CS4ALL effort is framed in terms of computer science and careers, there is room for more. It does not exclusively focus on middle and high schools, but says that students in elementary school should also have a computer science experience, and that in addition to standalone implementations, it should be incorporated into other content areas, such as science, math, or art. It recognizes that many schools are integrating computing education into the regular school day, and that an understanding of how to do this effectively could have wide-ranging benefits. It also builds on the rising demand from parents, more than 90 percent of whom consider computer science equally or more important than core subjects, with the highest demand coming from black and Hispanic families.

We see an opportunity to impact thousands of students and teachers by investing in the systems that will support educators to effectively deliver computing education to all students.

The Robin Hood Learning + Technology Fund seeks to invest in pre-service and in-service training toward the creation of a supported pipeline of computational thinking-ready elementary teachers and schools. Schools serving low-income elementary students in New York City will serve as demonstrations for informing and refining the models, as well as contribute to our collective understanding of integrated computational thinking. Ultimately, our goal is to identify the models with the greatest potential, and then scale these to transform learning for thousands of students. The Fund will invest in deep qualitative and quantitative studies to help build the research base and inform and inspire others to replicate the promising practices we uncover.

The Fund looks to the International Society of Technology in Education (ISTE) and the Computer Science Teachers Association (CSTA) to define the term computational thinking as a problem-solving process that includes:

Formulating problems in a way that enables us to use a computer and other tools to help solve them; logically organizing and analyzing data; representing data through abstractions such as models and simulations; automating solutions through algorithmic thinking (a series of ordered steps); identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combinations of steps and resources; and generalizing and transferring this process to a wide variety of problems.

Students proficient in computational thinking skills:

- **Know Seven Key Concepts** (Sequences, Loops, Parallelism, Events, Conditionals, Operators, and Data);

- **Can Apply Four Practices** (Experimenting and Implementing; Testing and Debugging; Reusing and Remixing; and Abstracting and Modularizing); and

- **Have Shifted Their Perspectives** around how they express themselves, ask questions, and approach collaboration.
WHY COMPUTATIONAL THINKING AND NOT CODING?

The world has grabbed on to the importance of teaching kids computing. The media serves us daily reports on the number of STEM jobs we will not be able to fill if we do not teach our kids to code and fills us with fear at the prospect of a technology-driven economy that will only have room for computer scientists. Although it is true that technology is changing our world, what we need is not a never-ending pipeline of programmers, but instead a generation of people with the power to invent, design and think creatively, and do so using all the tools at their disposal, including computers.

Although being able to code could directly benefit young people and lead to a job, it is unclear that a narrow focus on coding at the elementary school level will translate so directly. First, the nature and number of future jobs is highly uncertain. Second, programming languages will undoubtedly continue to change, making learning a specific language without a broader focus on developing a robust skillset that transfers to other domains unlikely to be useful a decade or more later. To paraphrase Mitch Resnick and David Siegel, if coding is going to make a true difference in children's lives, it is important to move beyond the traditional view of coding as simply a technical skill or a pipeline to getting a technical job. Instead, we must view coding as a new type of literacy and personal expression, a way for people to organize, express, and share their ideas, much like learning to write.

It is time for us to move beyond coding and toward computational thinking. As Jeannette Wing states in her seminal article on the topic:

Computational thinking means more than being able to program a computer. It requires thinking at multiple levels of abstraction. It is about conceptualizing, and not just programming. It is a way of thinking, learning and creating to solve problems, not a rote skill. It combines mathematical and engineering thinking and requires effective communication and collaboration. It is about generating and articulating ideas, not just knowledge of software and machines. It is about making meaning of the world we live in, not understanding computers.

Taking advantage of the full potential of computational thinking requires us to shift not only our thinking but also – and more importantly – our practices, to ensure that computing education focuses on leveraging computers to advance learning and thinking broadly. We assert that integrating computational thinking across the curriculum, and starting at a young age, is one strategy for achieving this.

WHAT PREPARING TEACHERS TO INTEGRATE COMPUTATIONAL THINKING LOOKS LIKE

Despite the growing interest in bringing computing education and computational thinking into our classrooms, we still have little clarity on how best to prepare teachers to do so.

Currently, most efforts to prepare teachers to integrate computational thinking have focused on training teachers once in the classroom, as opposed to while they are studying to become teachers. While some efforts, such as CS4All in New York City, aim to build teachers’ capacity generally, the majority are oriented around a single course, exam or product. The result is a continuous stream of teachers entering the classroom without the fundamental grounding to identify where, when and how to bring diverse computing tools and activities into their classrooms.

The Fund seeks to address these challenges by partnering with organizations and institutions to identify high-impact, scalable models for training aspiring teachers to use computational thinking as part of their practice. While colleges of education are just starting to explore how to do this, we do have some promising examples from states around the country. These include: schools updating their core curricula to include computing education in their methods courses; cross-departmental collaborations with computer science; revamping legacy technology courses; field placements in computer-science rich classrooms; and restructuring certifications to include mandatory digital literacy and computer science skills.

We also seek to collaborate with organizations to build the computing education capacity of teachers broadly once they are in the classroom. There are nascent efforts to replicate the successful literacy and math coach model to train computational thinking coaches who can directly support teachers. We are also seeing emergent professional learning communities and systems of peer-mentoring. These are exciting first steps, that will hopefully lead us to a strong understanding of what dosage, point of entry, types of support, and methods, will yield the greatest results for students.
WHAT INTEGRATED COMPUTATIONAL THINKING LOOKS LIKE IN ELEMENTARY SCHOOLS

No one really knows what integrating computational thinking throughout the K–5 curriculum “should” look like, or how teachers would implement it. The Fund seeks to address these challenges by preparing elementary teachers to integrate and teach computational thinking and providing them the ongoing support to develop their practice once in a classroom. We intend to ground these efforts in a small number of elementary schools to learn together what yields the greatest impact for students and create a deep qualitative and quantitative research base to inform our work and the field.

But for all that we do not know, we are beginning to identify some of the required elements. Dr. Irene Lee, in an article for the Association for Computing Mastery, provides three concrete steps for educators seeking to understand how to develop computational thinking among young learners, with an emphasis on the critical importance of moving youth from consumers to producers:

1. **THE USE OF RICH COMPUTATIONAL ENVIRONMENTS** in which students get under the hood to see, explore and customize the underlying abstractions and mechanisms, moving from user to creator.

2. **FOLLOWING THE THREE-STAGE PROGRESSION CALLED USE-MODIFY-CREATE**, which describes a pattern that starts with students using someone else’s creation, such as playing a ready-made computer game; then moving to modify the model, such as changing the character’s behavior; and finally creating their own digital artifacts and tools.

3. **LEVERAGING MULTIPLE DOMAINS TO INTRODUCE COMPUTATIONAL THINKING**, including modeling and simulation, robotics, and game design and development, among others.

Dr. Leigh Ann DeLyser of Computer Science for all NYC students (CSNYC), reminds us that the integration of computational thinking exists along a continuum that can be understood to move from learning experiences that focus primarily on developing computing knowledge (Low Integration), to experiences that effectively advance both computing and content knowledge through their interaction (High Integration).

<table>
<thead>
<tr>
<th>LOW INTEGRATION</th>
<th>MIDDLE INTEGRATION</th>
<th>HIGH INTEGRATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview: Stand-alone computing class that includes, builds on, or makes reference to core content from elsewhere in the curriculum.</td>
<td>Overview: Computational thinking skills and approaches used within a content subject to demonstrate knowledge.</td>
<td>Overview: Both content and computing teachers using a computational thinking and computing frame to enhance content learning and the development of computational thinking and computing skills.</td>
</tr>
<tr>
<td>Example: A once-a-week computer science class where students create a digital game around historical figures they are learning about in social studies.</td>
<td>Example: A social studies teacher asks her students to use a digital tool to create an interactive presentation about historical figures.</td>
<td>Example: A social studies teacher asks her students to first classify different historical figures based on their position on a given issue, and then create a Venn diagram showing trends, concentrations and overlaps in the data to make sense of historical events.</td>
</tr>
</tbody>
</table>
We understand that organizations and their school partners will be at different places along this continuum at different times and may use a combination of integration approaches that leverage both plugged and unplugged activities, and integrated and stand-alone classes. Our goal is for our partners to be aware of the possibility of high integration and strive toward it as a powerful and efficient model with the potential to enhance the existing curriculum and contribute to advancing student learning and achievement in an engaging and relevant manner.

CONCLUSION

The Robin Hood Learning + Technology Fund is dedicated to collaborating with organizations, school leaders, educators, and researchers to build the research base for integrating computational thinking and computing education in the elementary school setting. Our goal is to provide the models, evidence and tools needed to trigger a shift in practice with the potential to bridge the digital opportunity divide preventing many of our youth from realizing their potential.

ABOUT THE FUND

The Robin Hood Learning + Technology Fund is dedicated to unlocking the potential of technology to transform learning and advance achievement for low-income students in New York City. A collaboration among Robin Hood, the Overdeck Family Foundation and Siegel Family Endowment, the Fund focuses on seedung and uncovering bright spots of innovation in two targeted strategies for 1) developing elementary students’ computational thinking skills and 2) boosting literacy through a blended, content-rich approach. Across both focus areas, we seek to build the research base and share learnings with the field to support the replication and scale of promising approaches.

The Fund is guided by a uniquely experienced advisory board: John Overdeck and David Siegel, Fund co-chairs and the founders and chairman of Two Sigma; Laura Arrillaga-Andreessen, founder and president, Laura Arrillaga-Andreessen Foundation; Michael Horn, chief strategy officer, Entangled Solutions; and David Saltzman, co-founder and board member, Robin Hood and senior vice president, Two Sigma.

ORGANIZATIONS AND SCHOOLS INTERESTED IN PARTNERING WITH THE FUND OR LEARNING MORE SHOULD CONTACT US AT LEARNANDOTECH@ROBINHOOD.ORG