New York Hall of Science:
Computational Thinking School Strategy Guide

Table of Contents

Computational Thinking School Strategy Guide

This Computational Thinking School Strategy Guide provides information and resources on how elementary and middle schools can integrate the core constructs of computational thinking into their classroom curriculum. NYSCI's innovative approach to integrating computational thinking into classrooms engages students in practices of higher order thinking, metacognition and problem solving that sets them up for success across subject areas, and prepares them with decision-making skills for life beyond the classroom.

The guide is based on innovative professional development models for elementary grade teachers, developed by the New York Hall of Science and Public School 13 (Clement C. Moore) in Queens, NY, and funded by the Robin Hood Foundation Learning & Tech Fund. NYSCI has delivered this professional development to hundreds of K–8th grade teachers in elementary and middle schools across New York City's District 24, helping teachers to integrate computational thinking into their teaching and build their capacity to recognize and support those new practices within the context of their existing priorities and instructional constraints. This guide synthesizes what we have learned from those professional development efforts, and can serve as a road map for teachers as they take foundational steps towards integrating computational thinking across their curriculum, using both unplugged (without technology) and plugged (with technology) activities.

What Is Computational Thinking?

There are numerous definitions of Computational Thinking (CT) that can be quite confusing or complicated. A modified definition that resonated with our project team is: *Computational thinking is a problem-solving strategy that is derived from Computer Science but is also applicable in any domain. This strategy includes the following core constructs:*

Decomposition —
Breaking a problem into smaller, more manageable parts.

Pattern Recognition —
The ability to use prior knowledge to find patterns within the smaller problem that will help solve the complex problem more efficiently.

Abstraction —
Removing unnecessary information and focusing on what is truly important in a given situation.

Algorithm/Debugging —
Developing a series of instructions to solve the original problem, and evaluating the solution to address any errors.

Although additional core constructs are sometimes included in definitions of computational thinking, we view these concepts as problem-solving strategies that could be applied towards any subject matter with or without the use of technology. These core constructs are not always implemented in a linear or step by step process: rather, they can be drawn on to respond to the needs of the problem at hand.

As you begin to plan for integrating computational thinking into your existing lessons, you may notice that you may already encourage students to use some of these problem-solving skills. The benefit of the broader framework of computational thinking is its emphasis on combining these strategies, which all support your students' metacognition — their ability to think critically about the nature of the problem they are trying to solve.

"When you think of CT, a lot of people are talking about tech and I think the important thing about CT is that it's a way of critically thinking and you can apply it to any situation that you come across."

— 5th Grade ELA Teacher

Think of computational thinking as a coordinated toolkit of problem-solving strategies — as your students become familiar with them, and how they can work together, their ability to analyze and break down complex problems will grow.

Why Integrate Computational Thinking?

With the development of explicit Computational Thinking integration strategies, the push for CT into the classroom has grown tremendously. Educators, researchers, and policymakers are realizing more and more how valuable understanding and integrating CT can be, for a multitude of reasons. Two particular reasons that have resonated with the teachers and school administrators that we have connected with include:

- CT is a problem solving strategy that spans across all subjects and can help students of all grades and abilities to tackle complex challenges, with or without the use of technology.
- CT strategies can help develop students' perseverance across multiple school subjects, and is also applicable to problem solving in everyday life.

We have all had the experience of watching a student struggle with a problem or challenge. Often, we know the student actually has the skills and knowledge they need to solve it, but some aspect of the problem or its context is preventing them from recognizing how their knowledge can be matched to the task at hand. These are moments when metacognition — our ability to think about our thinking — is critical. Integrating computational thinking into everyday instruction can help our students by giving them tools that they can use to break down problems, analyze them, and recognize solution pathways. These kinds of support, which emphasize helping students with the process of demonstrating what they know and can do, is just as critical to their long-term success as engaging them with new content and new skills.

Familiarity with CT can also benefit teachers, encouraging them to integrate multiple problem-solving strategies into content-area lessons to support their students' success. For example, imagine a classroom struggling with math problems that focus on adding and subtracting decimals. The students are comfortable with actually doing the math but constantly put the decimal point in the wrong place. A computational thinking approach to this issue might have the teacher develop an activity solely on having students write out an "algorithm" on what their steps would be to solve these types of problems. This approach helps students hone in on what their process will be and also provides opportunities for students to share their process to learn from one another.

"My daughter explained to me that Computational Thinking is something we can apply to everyday life and is a way of simplifying problems."

— Mother of a 5th Grade Student

Computational Thinking Examples

Below is a chart that demonstrates the computational thinking process and how it can be applied to everyday tasks or problems.

Everyday Life Example	Decomposition	Pattern Recognition	Abstraction	Algorithm/ Debugging
Complex Task: Clean up a messy room	The floor has to be cleaned, my bed has to be made, and the trash has to be emptied. *The complex task has been broken down into smaller, more manageable tasks.*	Clean clothes go in the dresser. Dirty clothes go in the hamper. *Recognizing how to sort clothes based on patterns or prior experience enables you to accomplish the task more efficiently and reduces errors.*	It doesn't matter what order my books are in as long as they are neatly placed on the shelf. *Focus on the factors that are important to know or to help eliminate unnecessary steps.*	Pick up clean clothes and put them in a closet or dresser. Pick up dirty clothes and put them in the hamper. Sweep the floor, use a dustpan to pick up dirt, and place it in the trash. Throw out the trash and place a new garbage bag in the trash can. Check over everything to ensure that it is clean. *A series of steps to take that optimizes the speed and efficiency of completing the task. This algorithm can also be used to address similar problems.*

Whether you are brushing your teeth, preparing a sandwich, or fixing a tire, you are constantly utilizing your own computational thinking skills to accomplish tasks throughout your day. As an adult, you've developed these strategies over time — so you know how to organize a trip to the grocery store, or how to decide on an approach to cleaning out the garage.

Similarly, our students may already use these CT strategies at school in various subjects. But these strategies may be more accessible for some students than for others — and for all of them, they are unlikely to be explicitly aware of when and how they activate these strategies to support their work. The opportunity we give learners when we teach computational thinking strategies explicitly is that they can gain awareness of them, recognize when they are useful, begin to use them in combination, and access them to support their work over time.

Perhaps one of your students may spontaneously use a simple algorithm to solve a series of multiplication problems in math class. Another student might decompose a large amount of information into sections of an argument, as they create an essay on World War I for history class. But either of these students might be stumped in another setting, or with content they are less confident about, and not recognize that they can transfer these problem-solving strategies to new settings. By teaching computational thinking explicitly, we help them name and know these tools, which can help them to tackle novel and increasingly complex problems over time.

Constructing a model of a butterfly life cycle for science class, or drawing a landscape scene for art class — students can use computational thinking strategies as they work towards accomplishing these tasks.

Subject/ Assignment	Decomposition	Pattern Recognition	Abstraction	Algorithm/ Debugging
Math — Solve 6(1+2)	What are the operations being used here and which should I start with first?	What are the set of rules that should be applied to solve this problem?	What are math properties that I don't need to be concerned with?	Solve what is in the parenthesis first. Then multiply by 6. Double check answer.
History/ELA — World War I essay	What are the parts of an essay?	What resources can I use to research information for my essay based on past experience (library, internet, etc.)?	What information is irrelevant to include in the essay?	Write the topic introduction paragraph. Write the main body of the essay to support the main point. Write the conclusion to connect all points together. Double check facts, references and grammar.
Science — Design butterfly life cycle model	What are the lifecycle stages and what are the materials that I need or have to create the model?	What do I already know about using these materials?	What specific criteria did the teacher give me for the model?	Research stages of life cycle. Draw a circle diagram. Add pictures of each stage and label. Color in model. Double check for accuracy and meeting criteria.

Integrating Computational Thinking at Your School

1. Familiarize Yourself With Computational Thinking

Getting started with any new approach to designing and implementing lessons can be very challenging so we recommend starting with familiarizing yourself with examples or resources that are relevant to your work and are easier to understand. This step proved to be beneficial for the project team and helped begin to lay a foundation of understanding around CT.

Below are some resources that we recommend to help get you started:

Computational Thinking by JULES

This animated video provides a very manageable definition of CT and its core constructs. It also showcases an example of applying a CT approach to fixing a flat tire.

https://www.youtube.com/watch?v=mUXo-S7gzds&t=2s

Six Research Takeaways to Help You Understand Computational Thinking by Tech-Based Teaching Editor

https://medium.com/tech-based-teaching/six-research-takeaways-to-help-you-understand-computational-thinking-53233d0001a8

Research Notebook: Computational Thinking, What and Why? By Jeannette M. Wing

http://people.cs.vt.edu/~kafura/CS6604/Papers/CT-What-And-Why.pdf

2. Dive Deeper With Professional Learning Opportunities

Participating in professional learning experiences, especially with your fellow teachers and administrators, will provide you with more in depth tools and ideas for integrating computational thinking. Check with your school representatives, local or national professional learning providers (including science centers such as the New York Hall of Science) for computational thinking workshops that align with your needs and interests.

3. Introduce the Computational Thinking Approach to Your Students

It is important to build a foundational understanding with students on the CT approach in ways that students can relate to and are meaningful to them. Beginning early in the school year to review the CT approach through unplugged methods helps students understand the process first and how they can apply it to their daily lives and in school. Focusing on CT integration with unplugged activities also helps students transition the CT process to plugged activities once technology based activities are phased into their class curricula. Below are a few recommended methods that were successful at our original pilot school of PS 13:

- For early learners (K–1st), focus on building positive associations and familiarity with CT in relationship with their regular set of activities at home, school and during play. For example, introducing the skill of algorithms can be practiced throughout different routines done during the day such as asking students to share with you what are the steps taken to properly wash their hands when they are learning that skill after an art activity. If you think your students can handle the vocabulary, you can slowly introduce it to them with a similar approach listed above. Look for continuous opportunities to practice foundational CT skills within the structure of your normal classroom day or within your lessons targeting early childhood learning skills such as sequencing, sorting, pattern identification, etc.

- Demonstrate to students how to use the CT approach to tackle a complex task that students can identify with on a daily basis (ex. cleaning a messy room). Providing everyday examples and utilization of the CT approach will create a pathway for students on how to use CT approach in school. If you need to make specific accommodations for your students, you might want to consider focusing on introducing each CT component gradually so they have the opportunity to practice that one skill at a time.
- Display CT vocabulary charts or posters throughout your classroom or school. These provide constant reference points and reminders for students. Teachers can also develop their own posters or charts that connect CT vocabulary to specific classroom activities or daily tasks.
- Incorporate the CT questioning strategies into your daily classroom facilitation (regardless of class subject) to reinforce a computational thinking mindset with your students. This is very important to maximize the benefit of using CT as it gives students the opportunity to reflect and articulate their thought process on solving problems. Use CT questions below when conducting an activity or assignment in class to help guide students through the CT process and build a CT growth mindset.

CT Core Constructs	CT Facilitation Questions
Decomposition	*How can you break this problem down into smaller steps?*
Pattern Recognition	*What patterns do you notice that may be helpful in solving this problem? What do you already know about this subject that will help?*
Abstraction	*What are the important and unimportant parts to know to solve the problem?*
Algorithm / Debugging	*What are the specific steps that you need to take to solve this problem? How do you test it out to make sure these steps are correct and fix it if needed?*

4. Identify and Integrate CT Into Your Lessons

When integrating CT into your lessons, the goal is to provide additional support for your students so that they are all able to address the learning goals and objectives that you want them to accomplish. Computational Thinking may not lend itself to every lesson, so it is important to decide how and when it can best serve your students. If a challenge or problem within a lesson is relatively simple and you feel that your students will not have many issues with it based on your experience teaching that topic, then you should not look to introduce CT into the lesson. If the challenge or problem within an activity is more complex, multi-layered and/or that your students struggle learning that topic, then you may want to integrate CT.

Remember that CT is rooted in programming and computer science. So you might want to consider, what kinds of problems are computers good at solving, and what are they not so good at? Problems that require sorting, categorizing, tracking on information, or iterating on a process — these are all great candidates for computational strategies. Problems that are social or cultural, or creative and artistic challenges, might not be so well suited to computational approaches because they depend on personal experience and perspectives and creative, original expression of ideas and feelings. But there are no hard and fast rules here — if you see a way that a lesson might benefit from taking a computational approach, go ahead and give it a try!

The New York Hall of Science developed a Computational Thinking Lesson Planning Tool to help teachers identify where they see CT alignment in their existing lessons or to help teachers begin planning a new CT centered lesson. A blank template of the CT Lesson Planning Tool is available at the end of the guide for future use by teachers.

5. Incorporate Technology Into Your Classroom

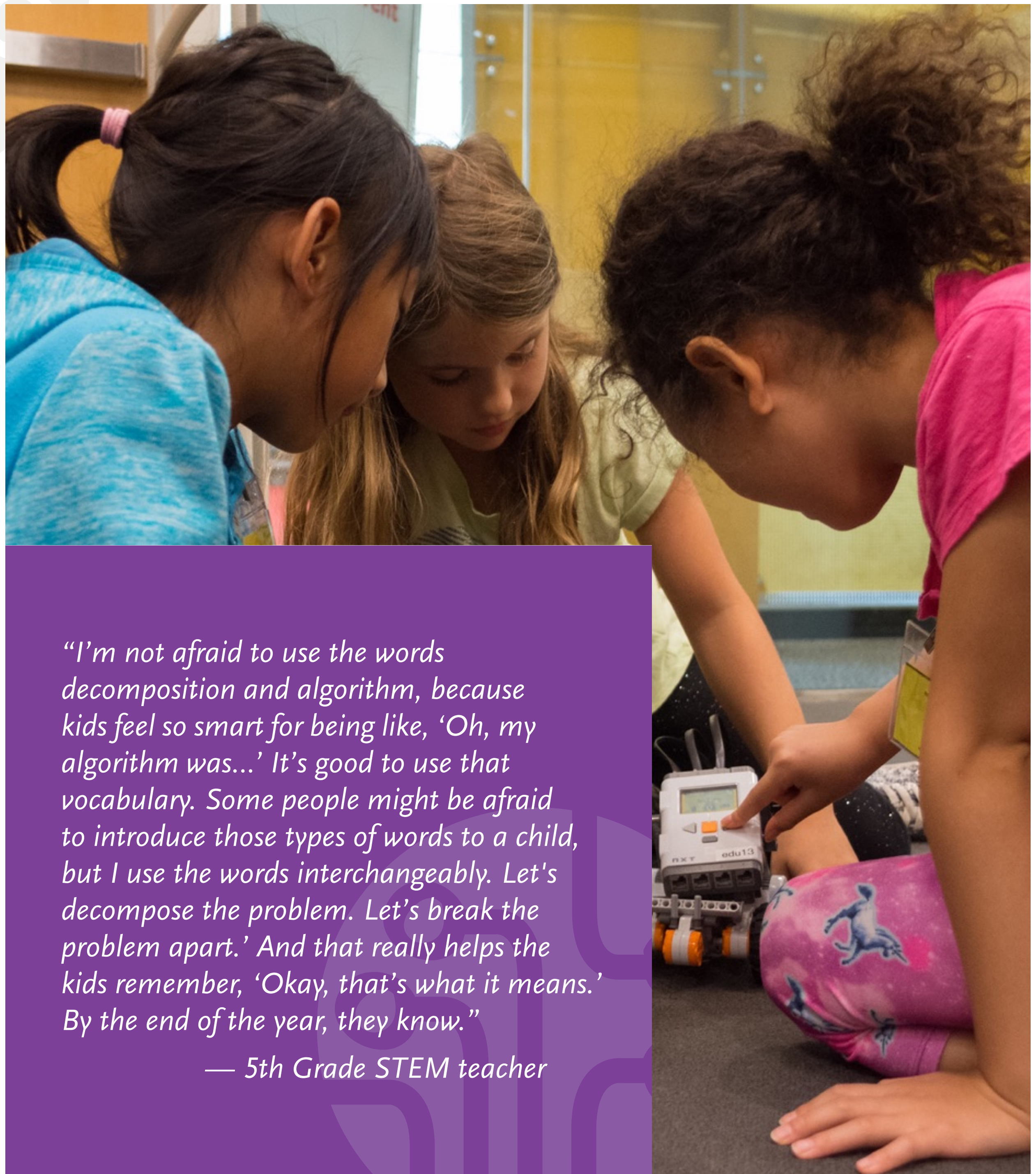
Thinking of ways to utilize technology within your lessons may be daunting due to a variety of reasons including time management, access to technology, or aligning to specific standards. However, investing time in planning how to incorporate technology into your class units will not only help build CT skills, it can also help your students be more engaged in the activity and better understand the subject material or content in your lesson.

Below are a few recommended tech tools.

Tech Tool & Online Resource	Recommended Grade	Activity Ideas
Sphero Edu.sphero.com	Grade 4 and up	*Math — Students navigate Sphero through a floor maze avoiding incorrect answers to a series of math problems.* *Art — Using a kiddie-sized pool with the inside floor covered in butcher paper and different spots of acrylic paint, students can program or use sphero basic navigation interface to create an art piece. Sphero must be covered in a protective case and plastic wrap.*
Code & Go blog.learningresources.com/coding/	K – Grade 3	*Math — Place pictures of geometric shapes in the maze and have students select matching shape cards and program the mouse to reach those shapes in the maze.*
Scratch Jr. scratchjr.org	K – Grade 3	*ELA — Create an animation that reflects narrative structure to a story.* *Math — Create a visual math story that demonstrates application of math.*
Scratch scratch.mit.edu/educators	Grade 4 and up	*Math — Design a game utilizing math operations blocks.* *ELA — Design an animation that demonstrates the theme or main points in a story.*
Botley http://blog.learningresources.com/wp-content/uploads/2018/03/Botleylessonplan.pdf	Grade 2 and up	*Science — Use Botley to investigate push and pulls on objects.* *ELA — Create a story based on Botley's obstacle course adventure.*

6. Pace Yourself!

It is important that teachers feel comfortable with how and when they are integrating CT that best suits their classroom needs as well as available resources. Computational thinking can be integrated in many different ways with different levels of intensity. Create a model for CT integration in your classroom or at your school that accounts for available resources, time management, and administrative support with opportunities to grow and expand. You might go completely unplugged, use your current lessons to build CT understanding among your students, and work toward developing plugged activities that involve students in applying their CT skills to coding or programming. The main goal is to use CT concepts, practices and perspectives explicitly, and in ways that are developmentally appropriate, to recognize, support, and discuss your students' learning.



"I'm not afraid to use the words decomposition and algorithm, because kids feel so smart for being like, 'Oh, my algorithm was...' It's good to use that vocabulary. Some people might be afraid to introduce those types of words to a child, but I use the words interchangeably. Let's decompose the problem. Let's break the problem apart.' And that really helps the kids remember, 'Okay, that's what it means.' By the end of the year, they know."

— 5th Grade STEM teacher

Additional Resources

The New York Hall of Science offers professional learning opportunities for teachers that provides training and support on computational thinking. Professional learning opportunities range from a one day workshop for foundational CT training to year long intermediate CT coaching programs for extensive training and lesson modeling within the school environment.

Computational Thinking Introductory Course

During this six hour workshop, participants learn the definition and fundamentals of Computational Thinking through lesson modeling and reflection of several unplugged activities. Participants will also receive training on how to use specific NYSCI developed tools that can help them brainstorm and monitor the progress of their own CT unplugged lesson.

Computational Thinking Unplugged Lesson Writing

During this six hour workshop, teachers will collaborate to design and share a Computational Thinking Unplugged lesson draft based on a grade appropriate lesson goal provided by a NYSCI instructor. Teachers will have the opportunity to have more individualized consultation sessions to address any specific needs or challenges with the audience they serve.

Computational Thinking Technology Introductory Course

During this six hour workshop, teachers will engage in a CT plugged based lesson while learning best practices for when and how to use tech in the classroom. Teachers will then be introduced and trained on how to use a grade appropriate tech tool through a design challenge activity.

Computational Thinking Plugged Lesson Writing

During this six hour workshop, teachers will collaborate to design and share a Computational Thinking Plugged lesson draft based on a grade appropriate tech tool and lesson goal provided by a NYSCI instructor. Teachers will have the opportunity to have more individualized consultation sessions to address any specific needs or challenges with the audience they serve.

Computational Thinking Coaching

In this school year long coaching model, NYSCI staff will work with school leaders to customize professional learning objectives and experiences for a cohort of teachers within your school. Through a combination of group workshops, individual coaching sessions, and classroom lesson modeling, teachers receive training and support to ensure CT integration meets the specific needs of their own unique classrooms.

For more information on these professional development programs, contact Anthony Negron, Director of Digital Programming for the New York Hall of Science, at anegron@nysci.org.

Computational Thinking Lesson Planning Tool

Use this tool to help guide how you incorporate computational thinking into your lesson.

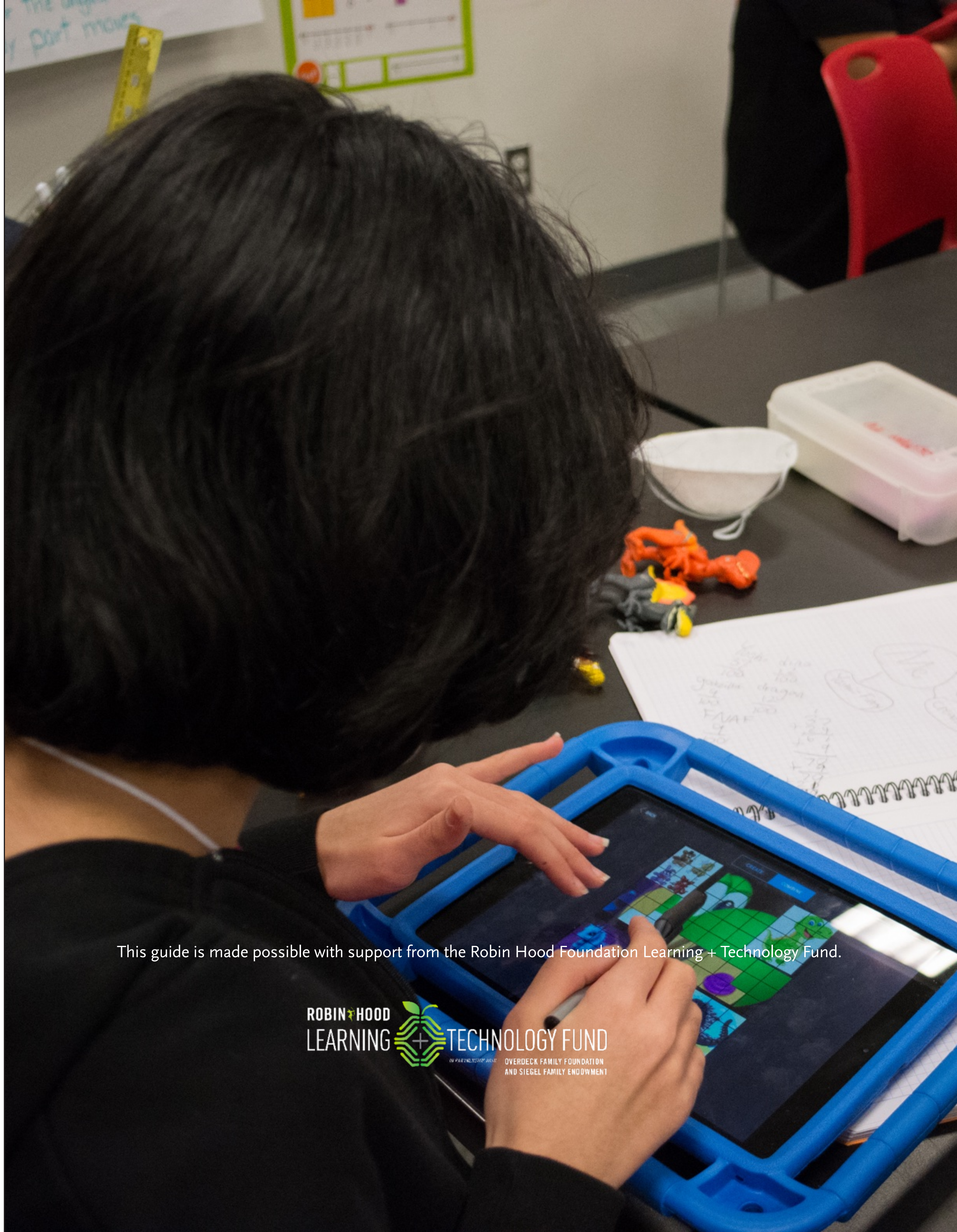
What are your lesson objectives? What is the main skill/problem/your students will be solving?

CT Process Below:	How is your lesson incorporating this CT item?
Decomposition: *Define a problem and break that problem into smaller, more manageable parts such as sub-problems, variables, categories, etc.*	**Does your main problem require the need to be broken into smaller problems? If yes, how so?**
Decomposition prompting questions	What questions will you use to help your students recognize how they use decomposition to solve this problem/challenge? Are there scaffolds that are needed to support students in this task?
Pattern Recognition: *Use prior experiences to find patterns within a problem.*	**What prior knowledge will be used? What patterns will they notice within the problem?**
Pattern Recognition prompting questions	What questions will you use to help your students recognize how they use pattern recognition to solve this problem/challenge? Are there scaffolds that are needed to support students in this task?
Abstraction: *Remove unnecessary information/ isolate critical information..*	**What extraneous details are you including in the problem?**
Abstraction prompting questions	What questions will you use to help your students recognize how they use abstraction to solve this problem/challenge? Are there scaffolds that are needed to support students in this task?
Algorithm: *A series of step by step instructions to solve a problem. Evaluate solutions to address any errors (debugging).*	**How will students use step by step instructions to come to a solution? How will students check and fix their work?**
Algorithm prompting questions	What questions will you use to help your students recognize how they use an algorithm to solve this problem/challenge? Are there scaffolds that are needed to support students in this task?

Computational Thinking Observation Tool

Use this tool to reflect on how well students engaged in computational thinking during your lesson and what modifications might be needed after implementing the lesson.

Observation Areas	Checklist	Evidence
Students Understanding of Content Area	☐ Are students using specific content area vocabulary? ☐ Are students correctly applying content area concepts to activity? ☐ Are they able to summarize what they learn?	
Students Understanding of Computational Thinking Concepts	☐ Are students describing computational thinking concepts (this may include using specific CT vocabulary)? ☐ Are they applying computational thinking concepts? Are they specifically grasping: ☐ decomposition ☐ pattern recognition, ☐ abstraction and ☐ algorithm? ☐ Are they debugging?	
Student Engagement	☐ Are students paying attention? ☐ Are they responding to questions? ☐ Are they asking questions? ☐ Are they productively collaborating?	
Activity Differentiation	☐ Are there modifications to the lesson for different learners in your classroom? ☐ Are those modifications being implemented with the students they were intended for?	



This guide is made possible with support from the Robin Hood Foundation Learning + Technology Fund.

ROBIN HOOD LEARNING + TECHNOLOGY FUND
OVERDECK FAMILY FOUNDATION AND SIEGEL FAMILY ENDOWMENT