Computational Thinking: Coding

Mary Bauernschmidt

Introduction

Common Core State Standards were adopted by Colonial School District in 2014[1]. The Common Core State Standards for Mathematical Practice outline the expertise that teachers should be working to develop, in students of all ages and academic abilities. In grade 3, instructional time should focus on four critical areas; multiplication and division; fractions; arrays/ area and two- dimensional shapes. In addition to the Common Core State Standards, there are a total of eight standards for mathematical practice from Common Core State Standards that teachers are expected to tie into their curriculum. These practices are the foundation for mathematical thinking and practice for students. These standards are to be thought of as a guide to creating more complex and absorbing learning experiences that can be applied to everyday life. They are as follows: make sense of problems and persevere in solving them, reason abstractly and quantitatively, construct viable arguments and critique the reasoning of others, model with mathematics, use appropriate tools strategically, and to look for and express regularity in repeated reasoning.

As an offshoot of the Common Core State Standards, the students in the Colonial School District take the Smarter Balanced Assessments. These tests are given in grades third through eight. The Smarter Balanced Assessments require students to apply knowledge and skills across multiple subjects to demonstrate critical thinking abilities. In mathematics, students perform multi-step problems and explain how they achieved their answers. For the 2017-2018 school year, 36.2% of third grade students scored a 3 or 4 on the math test and are performing on grade level. For the school year 2016-2017 the percentage was 47.1%. As far as student test scores/data go there is an obvious need for third grade students in Pleasantville Elementary to improve their mathematical critical thinking abilities.

It is my experience that students, both male and female, sometimes stereotype themselves as “not good” in math as early as the beginning of third grade. I have heard students say things like “I am never going to be good at math” or “This is too hard.” These students tend to consistently show task-avoidance during math instruction. This avoidance will lead to loss of instructional time and thus perpetuate their conception of their math skills as being low or none at all. At times, parents will validate these feelings by saying things like "Yup! She's just like I was in school; not a good math student." Through the use of computation thinking lessons, I would like to alter my students' belief system from one of self-doubt like" I will never be good in math" to one of hope "If I give effort, I can succeed in math." This change will help them to perform better in the classroom and on the state test: Smarter Balanced Assessment.

The eight standards of mathematical practice will be the focus of the unit. The unit will be designed for the start of the school year. This timing will be the most beneficial, as it is intended to boost student motivation and self-efficacy in mathematics that will promote more intensive learning in the common core state standards as the school year progresses.

I have learned that computational thinking equips one with critical thinking skills. They allow the learner to conceptualize, analyze, and solve complex problems. Often critical thinking leads problems beyond what the learner thought possible to solve independently. I want to delve into this phenomenon to provide this experience to my third-grade students. I hope to use computational thinking situations and problems to enable my students to be successful at tasks they never thought possible, I could essentially increase their confidence as a mathematician. Confidence would be built through the use of "steps" and routines.

My unit will include the use of coding activities and Code.org. This curriculum unit is a mix of online independent practice, unplugged group activities, and discussion. In addition to the lessons I have created, I will assign lessons and monitor progress.

This unit will be beneficial to all third-grade teachers across Delaware and beyond. Students in all schools whether public, public, private or charter in my opinion struggle with the same self-doubts. I am excited to write this unit with the hopes of teachers who implement the unit see the students eyes light up when they realize that yes they CAN DO math!

Why Code? Rationale

It is clear how technology is shaping the world. More and more jobs require at least some basic computer knowledge and computational thinking. As students move through school, it is expected that they know how to use a computer. It seems to me that with the direction our world is going, more and more expectations will be placed on students to have coding skills. These skills afford more opportunities for them as they enter into the real world and workforce. The lessons involving coding contain are cross-curricular: Computer Science, English and Language Arts, Science, and of course, Mathematics.

Students need to start somewhere. So why not third grade? In Delaware, third-grade is the first time students take the Smarter Balanced Assessments. What better way is there for students to learn computer skills needed to take the Smarter Balanced Assessment than through coding?

Common Core State Standards

CCSS.MATH.CONTENT.3.G.A.**:** Understand that shapes in different categories (e.g., rhombuses, rectangles, and others) may share attributes (e.g., having four sides), and that the shared attributes can define a larger category (e.g., quadrilaterals). Recognize rhombuses, rectangles, and squares as examples of quadrilaterals, and draw examples of quadrilaterals that do not belong to any of these subcategories.

1CCSS.MATH.CONTENT.3.MD.D.8**:** Understand that shapes in different categories (e.g., rhombuses, rectangles, and others) may share attributes (e.g., having four sides), and that the shared attributes can define a larger category (e.g., quadrilaterals). Recognize rhombuses, rectangles, and squares as examples of quadrilaterals, and draw examples of quadrilaterals that do not belong to any of these subcategories.

CCSS.MATH.CONTENT.3.G.A.2**:** Partition shapes into parts with equal areas. Express the area of each part as a unit fraction of the whole. *For example, partition a shape into 4 parts with equal area, and describe the area of each part as 1/4 of the area of the shape*.

Learning Objectives

- Students will be able to make sense of problems and persevere in solving them
- Students will be able to reason abstractly and quantitatively
- Students will be able to construct viable arguments and critique the reasoning of others

- Students will be able to model with mathematics
- Students will be able to use appropriate tools strategically
- Students will be able to attend to precision
- Students will be able to look for and make use of structure
- Students will be able to look for and express regularity in repeated reasoning

Essential Questions:

1. How can students learn to show persistence in the face of frustration?
2. What strategies will students use to build a load bearing structure?
3. How will students develop an algorithm?
4. What ways can students use the algorithm to encode it into a program?

Demographics

The Colonial School District is a grades Kindergarten through Twelve public school district located in New Castle, Delaware. My school is Pleasantville Elementary School. In my school, approximately 43% of students are African American, 22% of students are Hispanic, and 32% of students are White. Our English Language Learners population is about 21%, low income is 48% , and Special Education is 13% of the population. Pleasantville Elementary math curriculum is TERC Investigations. This curriculum was intentionally designed and sequenced to promote a deep understanding of mathematics. The units at each grade level represent a cohesive whole. Each unit builds on the previous unit across grade level strands and across grades. Investigations has been implemented in Colonial School District for at least ten years. One of the goals that guided the development of Investigations is to support students in order to make sense of mathematics and learn that they can be mathematical thinkers. This goal is one I hope to delve deeper into with my computational thinking unit in third grade.

The Colonial School District's mission is to maintain and expand a more rigorous teaching and learning environment that centers on providing all students with the skills necessary to be successful in the Twenty First Century. In our new grading system; Mastery Based Learning, which is a standards-based approach, our district has incorporated the mathematical practices by requiring teachers to rate student mastery. I believe my unit fits with this mission.

Introduction to code.org[2]

Code.org was founded in 2013. It is a non-profit and its' focus is to expand participation in computer science by making it available in more schools. One of its goals is to increase participation by girls and underrepresented students of all races. Their vision is that every student in every school should have the opportunity to learn computer science. Code.org believes computer science and computer programming should be a part of the core curriculum in education. Code.org curriculum and courses are free.

To get started using Code.org in the classroom, teachers need to set up teacher accounts. Then a class code will be generated. Students will need this code to log into their account. Each student's name and age will need to be inputted by the teacher. A secret word will be assigned to each student. Once this is completed students can begin working within Code.org, and their progress through the units will be saved. The tab "my dashboard" has student's progress.

I assumed introducing anything in a third-grade classroom with the word "coding" would be a challenge. I pushed myself to take on this challenge. And yes! I can do it. So, can every other teacher.

There are courses for students built into the Code.org website. You can read the descriptions and pick the one you think fits best for your students and their abilities. Since many of my students are not reading on grade level, I'm inexperienced with coding, and my main goal is to increase their self-perception of them as mathematicians, I decided to load my third graders into course one. So originally, Courses 1-4(CS Fundamentals International) were designed as a starting point to acquire the basics, no matter what the age of the student who was coming to the discipline. But Code.org went further and developed additional courses to meet the need of teachers asking for more content since many students across the country have been exposed to computer science content. And those courses are called CS Fundamentals A through F.

Course one contains eighteen lessons. A few are "unplugged." This means students are not working within Code.org but rather with the teacher and without computers. I chose to rework the unplugged "Building a Foundation" because it aligns directly with the original goal of my unit: to support students to make sense of mathematics and learn they can be mathematical thinkers. Create a sense of accomplishment and confident. And this is what the following lesson sets out to achieve. Third grade mathematical practices are a work in this lesson.

Activity 1: Unifix Cube Structure

In this lesson, the goal is for students to work together to build a structure that would be able to hold a pencil. This activity practices planning, building, cooperation, and reflection skills needed in coding. This is a precursor to the more complex lesson “building a foundation.”

Modifications to this lesson can be to ask students to build the tallest structure, the longest structure, and/or finally, to build a structure to hold a pencil. Structures can be changed to represent holiday themes, classroom needs, etc.

To start the lesson, gather your students on the rug. Challenge students to identify and give the students time to think about different shapes. Then have the students do a “turn and talk” telling their partner all of the shapes they came up with. Allow about two minutes for sharing. Once partners have shared, the call on students and record shapes on poster board. Next, ask the students where we see these shapes in the real world. Examples for each shape can be listed next to the shape. For example, rectangle; apartment buildings, octagon; stop sign, circle; plate. Discuss the importance of and influence of shapes when building structures in our environment.

For the activity, introduce the task. The task is to independently build the tallest structure using 10 unifix cubes within five minutes. The follow-up task is to build the longest structure they can within five minutes.

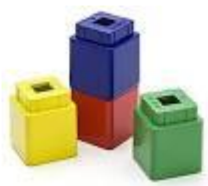
Next, assign partners and together the students will build a structure with their combined 20 cubes that can sustain holding a horizontal pencil. The timing for this would be 15 minutes. Teacher encourages students to make revisions and to persevere throughout the task.

During the building, the students would need to have their own supply of unifix cubes as well as their own individual work space. While they are working, give the students constructive feedback and words of encouragement.

After the time has elapsed, allow students time to walk around the room and view all of the structures making a mental note of what was successful and what was not. Then, the pairs have will draw their unified structure on the recording sheet and state what they would do differently if given 10 more unifix cubes. Students also will rate their own structure by success 3 worked great, 2 could have been better, 1 frustration and describe why.

Unifix Cube Structure Recording Sheet

Diagram of Structure



If I had the opportunity to do this acivity again with 10 more cubes I would ____

I feel like I ....

(circle one)

3- success!! I worked the best I could work!

2 - I could have been better.

1- I feel very frustrated and here's why...

____
____
____
____
____
____

Activity 2: Building a Foundation

The goal of the second activity is to engage students in a difficult challenge to highlight strategies for showing persistence in the face of frustration. This lesson teaches that failure is not the end of a journey, but rather a step towards success. The standards addressed are CCSS.MATH.CONTENT.3.G.A. and 1CCSS.MATH.CONTENT.3.MD.D.8.[3]

In this lesson from Code.org[3], students are asked to build a load-bearing structure using common materials. The structure will be tested on its ability to hold a textbook for more than ten seconds.

Prior to starting the lesson, gather students on a rug or in circle in chairs. Ask students to think of a time (closing eyes is optional) where they tried something new and didn't succeed. Give them examples like playing a sport, or a tough video game etc. As they are silently thinking, ask them to think about how that felt. Next have the students "turn and talk" to a student next to them. Make pairing arrangements if needed. Instruct the students to talk about the things they were feeling, thinking, saying and doing during this time of trying something new and not succeeding. Once the students are done sharing teacher will take four volunteers to share out their own experience.

During this lesson, students need 10 gumdrops, 20 toothpicks, and a ruler (with a partner). Students need to have a disposable surface on which to work, small paper plate, and/ or paper towels since sugar from gumdrops gets messy. In the provided lesson, students will build a structure that can hold a book for ten seconds.

I have created and added a “Gum Drop House” graphic organizer that needs to be reviewed and read aloud. Students will fill out “Gum Drop House” graphic organizer as they are completing the task.

To supplement the lesson, there are guiding questions to pose to groups of students such as Questions that provide hints/guidance during the building process: (use at your own discretion)?, *How will your roof affect the home’s ability to be tested?*, *How could you strength the joints?*, *Since you found that one triangle is good, would two be better?*, *How could you broaden the base to give the house more support?*, and *How does the number of toothpicks stuck into one gum drop affect the strength of the joints?*

As students work, it may be helpful for you to place some idea “examples” at the front of the room. Do not announce that they are there. Simply encourage students to take a walk if they get frustrated. Try to encourage students to locate the tips on their own if at all possible. This helps students feel like they “discovered” something that helped them, rather than being rescued. Make sure you also have “tips” for being persistent and for dealing with frustration written in a clear location.

Tips for Overcoming Frustration

- Count to 10
- Take deep breaths
- Journal about them
- Talk to a partner about them
- Ask for help

Tips for Being Persistent

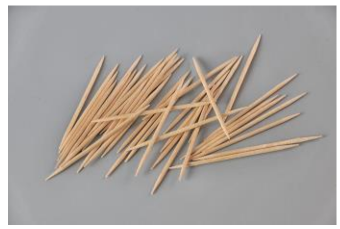
- Keep track of what you have already tried
- What is happening?
- What is supposed to happen?
- What does that tell you?
- Make a change and try again

To wrap-up the lesson, hold a Flash Chat. This Flash Chat is meant to help the students come to terms with the negative emotions that they felt during the project, and it will help them see that they were not alone in their struggle. It should also prepare them to recall the tools that they used to help themselves overcome frustration and be persistent so that they can access those tools again when solving other mathematical situations.

As student reflect on the activity as a class, using the following prompts as a start.

- Did anyone feel frustrated during this lesson?
  - Can you share what that felt like?
- (Alternatively, you can ask "Who felt frustrated at some point during this lesson?" and cheer for the students who raise their hand.)
  - Then YOU won this lesson, and YOU won this lesson, and YOU won, too!
- Was there a time that you thought about giving up?
  - What were you feeling, saying, doing, or thinking at that time?
  - How did you get past that feeling?
- Do you think that you would be more proud of yourself for solving something that was easy, or something that was very, very hard?

End the discussion by bringing the context back to the warm-up and question *Where would you be now if you had not been persistent when you were learning to walk as a baby? Or if you had been too frustrated to keep going when you were learning to talk? What other things did you learn to do, even though they were very, very hard?* In a pair, have students discuss this thought exercise with an elbow partner to come up with 1-3 things that fit that description. To share out, have students respond to the question on the back of their graphic organizer.

Gumdrop House Graphic Organizer

Using 10 gumdrops and 20 toothpicks make a house-like structure that can hold the weight of a heavy book. Record the house's height and width in the measurement column. Then describe what happened to the house after testing. If time permits try another shape.

I predict that the _______________-shaped house will be able to hold the weight of the book.

Shape	Measurement	Testing Results

1. What shape made the weakest home?

2. What shape made the strongest home?

3. If you had more gumdrops and toothpicks how would you make your home stronger?

Activity 3: Moving Forward

The next logical step for me after observing my students move their way through CS Fundamentals International course one was to take the leap and get my students into the course recommended for 3rd graders who have had some prior coding experience. For Code.org, this is the course they recently added after getting feedback from teachers wanting more rigorous lessons. This is CS Fundamentals course D. In this course students develop their understanding of algorithms, nested loops, while loops, conditionals, and events; as well as learning about digital citizenship. The context for the video is types of things that robots can do. The video is titled "Asimo By Honda."[4] This is about a four minute video. Discuss with the students; How do you suppose that robots know how to do the things that they do? Do they have brains that work the same way that ours do?

The goal of this quick discussion is to call out that while robots may seem to behave like people, they're actually responding only to their programming. Students will likely refer to robots from movies and TV that behave more like humans. The first lesson in Course D is graph paper programming. This is an unplugged lesson -so I have reworked it to be more user friendly for teachers.

Activity 4: Letter Graphing

The goal of this lesson would be for students to practice with recording and drawing on a grid in preparation of the up and coming "graph paper programming" lesson.

Students will get the opportunity to draw a grid, follow verbal directions by labeling the grid in certain ways, and practice recording steps others are to follow that end up creating their individual design.

As a whole class, distribute a sheet of plain white paper to each student. Instruct and model how to draw a grid of 4 (2x2 array). Teacher than gives verbal commands for the students to follow. They are listed below.

Step 1: Draw a circle in the center of the top left box.

Step 2: Draw a circle in the center of the top right box.

Step 3: Draw a star in the middle of the paper where all 4 boxes connect.

Step 4: Draw a letter U that spans from the middle of the bottom left box to the middle of the bottom right box.

Ask the students to write on their paper the name of the design they just created. Then, repeat the steps aloud on a dry erase board and dramatically reveals a smiley face!!

In the core of the lesson, all students will be given the opportunity to practice using the grid by drawing a letter of their choice. Distribute additional paper with a 2x2 grid of four preprinted. Tell the students to choose one of these letters - A E F H I W Y. Instruct the students to draw it on their grid. Next model how to record the steps by using Y as an example. State "If my letter is Y, I might write. 1) Bold the line between the bottom 2 boxes only. 2) Next, draw a diagonal line from the bottom left of the top box to the corner. 3) Draw a diagonal line from the bottom right of the top left box to the corner." Next, have students record the steps that they would tell others to take to draw that letter without revealing it.

Allow students time to choose their letters and to draw it on their grid. Have ample grids available for students who might need a fresh grid. Keep the modeling steps on the board and encourage students to use them as they record their own steps.

Try saying one or more of the following statements as you circulate and monitor the students work, "You're on the right track now", "That's coming along nicely!", "You've just about got it!", and "That's it!"

As students finish up writing their steps, pair them up. Provide each pair with new grids. Have the students verbally state their steps to their partner. Partners are to see if they can draw the letters following the directions. Then they switch roles. Teacher is to remind students that revisions to the steps might be needed. That is exactly what computer programmers do when designing programs in the real world.

Activity 4: Graph Paper Programming

The goal of this Code.org activity[5] is to build critical thinking skills and excitement for the course. In this lesson, students will learn how to develop an algorithm and encode it into a program. Standards addressed CCSS.MATH.CONTENT.3.G.A.2.[6]

By "programming" one another to draw pictures, students get an opportunity to experience some of the core concepts of programming in a fun and accessible way. The class will start by having students use symbols to instruct each other to color squares on graph paper in an effort to reproduce an existing picture. If there is time, the lesson can conclude with images that the students create themselves.

Again, watch the video: “Asimo by Honda”[7.] This will give students the cans. Push them to consider robots that they've seen or heard of in real life, like Roombas, or even digital assistants like Amazon Alexa.

It is the teacher’s choice to have open discussion or have students respond in writing to the questions ahead of the discussion.

In this activity, students will get to program robots...and they're already here in the room! It's you! Tell students “We're going to write programs using symbols with special meanings to help each other recreate a picture. First, we'll practice together as if I am the robot and you are the programmers, then we can break up into groups so that everyone can get a turn. If you want the computer to do the list of steps you have to program it. A code the machine knows how to read. Say “I’m going to turn into a robot. I only follow specific instructions. The goal is to fill in this graph so it looks like a checkerboard or every other square is filled in.” Teacher draws below image on dry erase board or poster.

Tell the class that their job as a class is to give me instructions. We will do this as a class. I have a very specific set of commands. Continue with Code.org lesson.[8] As the class gives you directions and moves through the lesson, record their steps on the board for future reference. At the end ask “What could we do to make the writing not so laborious or difficult?” “I mean my hand is really tired.” Get some student responses. Then say “Could we draw a simple to represent the command? “Go through each original robot command symbols and talk the students through creating symbols for the appropriate commands in sequence. “Now if we were to do another one with a different pattern it would be a lot less writing.” A prepared set of symbols is available, or your class can create a unique and efficient set of symbols.

At the end of the lesson, the grid should look like a checkerboard; however, you can modify the activity to make different designs and patterns.

After completing the prescribed lesson, divide students into pairs or small groups. Have each pair choose an image to recreate, discuss and convert an algorithm to draw, and trade with other groups to try.

As students are working in pairs or small groups teacher may say things like “Terrific Teamwork!”, “One more time and you’ll have it”, “Keep Trying”, or “Make a change and try again.”

To wrap-up, have students write about what they learned, why it’s useful, and how they feel about it can help solidify any knowledge they obtained today.

Journal Prompts (higher order thinking questions): What was today’s lesson about? How did you feel during today’s lesson? Draw another image that you could code. Can you write the program that goes with this drawing? What other types of robots could we program if we changed what the arrows meant?

What I love about coding!

In order for the teacher to be able to evaluate student’s excitement and learning highlights, teachers can take students through a maître-de sharing experience. Students are asked to record three statements. These statements could be things they have learned throughout the coding lessons, they could be statements about which lessons they liked the best and why, or they could be how they plan on continuing their learning on coding. Once students have written these out on paper, the teacher then rings a bell. Students are instructed to get into groups or “tables” of three. Then each student shares out one of their statements while the other students stay quiet and listen attentively. Once all students at the “table” have shared out, the teacher rings a bell and students move into a new table of three and the same procedure is acted out.

It is important for elementary students to understand and be able to work with the technology around them. Coding at an early age prepares them for the future. It helps with communication, creativity, math, writing, and building confidence. This unit accomplishes all that while at the same time while building excitement and positive energy for school in general.

Bibliography and Teacher Resources:

"Developing Computational Thinking and Coding Self-Esteem." Learning.com, 15 June 2017, www.learning.com/news/developing-computational-thinking-and-coding-self-esteem.

"Grade 3." Common Core State Standards Initiative. Accessed February 12, 2019. http://www.corestandards.org/.

Pólya, George. *How to Solve It: A New Aspect of Mathematical Method*. Princeton Publishing: Stellar Classics, 2013.

"What Will You Create?" Code.org. Accessed February 12, 2019. https://code.org/.

Vaidyanathan, Sheena. "What's the Difference Between Coding and Computational Thinking? - EdSurgeNews." *EdSurge*, EdSurge, 23 July 2018,

Notes

[1]Common Core State Standards, http://www.corestandards.org/Math/Content/3/introduction/
[2]Coding.org, https://support.code.org/hc/en-us
[3]CCSS, http://www.corestandards.org/Math/Content/3/introduction/
[4]Coding.org
[5]"Asimo" Honda, https://www.youtube.com/results?search_query=asimo+robot
[6]Coding.org
[7]CCSS
[8]"Asimo" Honda
[9]Coding.org