I completed my first paper on computational thinking in 2020. The book "The Math(s) Fix" by Conrad Wolfram published in 2020 was a key text to guide me through that assignment. The book explored Wolfram's concerns with mathematics instruction, and ways that mathematics instruction needed to shift to let computers be the tools of computation. This would allow students to shift the majority of their thinking from calculation to problem solving, providing more opportunity for them to work on the development, deconstruction, and solutions to the problems they were looking at (Wolfram, 2020).

For this project, the key text has been Mindstorms by Seymour Papert originally published in 1980. Papert is considered the originator of the term computational thinking, and Mindstorms is a rather interesting read 43 years after it's original publication. Like Wolfram, Papert explored computer use in school, and how it had the potential to change how students approached mathematics and problem solving. He was invested in how education could ensure that the technology of the day was used to its full functionality (Papert, 2020). Many of the concerns raised in 1980 still relate to how we use, or how we should use technology in schools. Papert will come up again in a later episode or two.

If you listened to the introduction - thank you - This is episode 1: What is computational thinking? I'm Stephen Hadden

So let's get into some of the theoretical, shall we...What is computational thinking?

Computers and computing devices have altered how education can be delivered, and they have altered and continue to alter the way that we explore data and seek solutions to questions and problems. Jan Cuny, Larry Snyder and Jeannette Wing (as cited in Wing, 2010) provide this definition, “Computational Thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent”. If we can get computers to provide solutions to our problems, then it is probably a good idea for all of us to think about how computers work and how computers arrive at the conclusions that they do. It is not about understanding how computers think (because they do not think without input), but more about how we think, how we use our devices to make things better or solve things that need solving (Vegas et al., 2021).

Jeannette Wing kicked off the popularization of the term computational thinking and set the direction for others to develop definitions and carry out their explorations . She determined that the foundations of computational thinking is human thinking, that it is about problem solving, and that it is not necessarily about programming computers, but the processes to work toward solutions to problems or tasks using what tools make sense (Wing, 2006). Others have continued to explore the ideas and applications of computational thinking. Educators and scholars have proposed a number of concepts and practices that relate to computational thinking, and how to explore computational thinking within our school contexts.

Grover and Pea (Lee et al., 2020) provide a collection of concepts and practices that encapsulate the main concepts of computational thinking. I could have used a number of other authors - Barr and Stephenson, Palts and Pedaste, Csizmadia et al. -

but at this point, lets go with one framework and work from that. As I list the concepts and practices - I will provide a very short application or thought on each, but I want you to think about how those practices and concepts might already be present in your instruction, and in the curriculum that you use. I will try to use a variety of experiences across many grades, and we will explore in later episodes why computational thinking fits with instruction including elementary instruction, and how we might explore it in elementary and middle years grades.

Shall we make this a little fun? - kind of like a top ten list (well top six and top five). According to Grover and Pea, here are the Top Six Concepts of Computational Thinking (In no particular order)

1. Logic and logical thinking

    - We see this in every math curriculum - this relates to the ideas of connecting our thinking to new topics. There are plenty of games, board games and logic puzzles that we use to explicitly explore this concept.

    - There are plenty of ELA and social connections with debate, structured writing, exploring problems and developing solutions.

    - This is covered explicitly in computer programming through boolean logic (AND, OR, NOT).

2. Algorithms and algorithmic thinking

    - Very common in food studies, and most practical and applied arts - following plans, creating plans, and revising plans.

    - Included in mathematics instruction, the scientific method, and the structures for writing sentences, paragraphs and essays.

    - According Grover and Pea, there are three building blocks in algorithmic thinking: sequence, selection, and repetition.

3. Patterns and pattern recognition

    - This is my favourite thing to point out with my grade 3 mathematics class. Where are the patterns in this list of numbers? They love it!

- Patterns in data help us analyze historical information, methods of speech in poetry, and sections of music. The possibilities are endless.

4. Abstraction and generalization

   - We have to abstract ideas and create generalizations all the time. Abstraction is considered information hiding. Say we need to calculate how many cookies are on the table, and we have 4 cookies on each of 3 plates. We will abstract the cookies and plates to numbers and signs so that we multiply 4 by 3 to get 12. So the concrete or physical description is gone, leaving the mathematical phrase. Generalization is when learn that the abstract phrase 4 x 3 = 12 works for counting cookies and we can also use that to figure out 4 x 3 for anything!

5. Evaluation

   - test and redo

   - We have lots of fun examples, of course as teachers we have summative assessments to complete, but we also provide plenty of formative assessment, and students can self-evaluate their progress.

   - "Hey, Jon, let's go and add a little more detail to those sentences."

   - "Uh, oh, looks like your tower didn't withstand the earthquake test - what shapes could you include that might make for a stronger structure?"

   - "Mr. Hadden, - this code isn't working", "Oh, it looks like you might have a few too many blocks - did you try running it step by step?"

6. Automation

   - This is about getting something else to do the work for us. This one may actually be hard to consider apart from technology. Though automation can use other machines to automate tasks.

   - With current technology, we have to consider whether tools like ChatGPT and other Artificial Intelligence tools are beneficial not because they can work, but because of how they work. We have to evaluate the automation and whether it allows us to think deeper, do stuff with more complexity, or does it just make us dumb and lazy.

   (Lee et al., 2020)

Those were the key concepts of computational thinking. There are also the practices of computational thinking. These are the things that we have to carry out to be doing computational thinking. You might hear these terms or descriptions and think - I already do that - and yes, you probably do. The only additional thing you may need to consider is, what tools can I use to do this, is there a way that using computer technology may help me get to an end product or solution easier? This is the thought that maybe doesn't run as easily through our minds at first.

Alright, five computational thinking practices in no particular order from Grover and Pea. CT practices include:

1. Problem decomposition

    - All subject areas are going to have problems that require breaking down. We may break it down into smaller topics, assigned tasks for group members, deal with one section at a time, or determine what tools we need for which parts. This is across all subjects and curricula.

2. Creating computational artifacts

    - This is going to involve a computer. But it's not just coding (which is a common misconception) What about a video instead of an essay, what about a graphic design poster instead of a drawn piece of art, what about using a spreadsheet to draw a graph -these are examples of computational artifacts

    - Coding is an important part of this as well. Creating data visualizations to study a science or social studies topic, and building tools to carry out calculations in math would be great artifacts that involve coding.

3. Testing and debugging

    - This relates to the concept of evaluation. We are going to make mistakes. In computational thinking mistakes are considered part of the process - they will happen, the important question is, what will we do to try to fix them?

    - Design thinking and engineering thinking processes - which include many similarities to computational thinking - contain cyclical processes to fix mistakes and are often used in STEM and STEAM challenges.

4. Iterative Refinement (incremental development)

    - This takes the process of testing a little further - Even it if works, can it be refined? There is the concept of elegance in programming, meaning that the code produced is able to do the task in a way that is shorter and requires less resources. Another aspect of incremental development is that there can be more than one way to get to the product.

5. Collaboration & Creativity

   - We explore opportunities for students to work together in all forms of education - the little kids like working together, and the senior students tolerate it.

   - Creativity takes on all sorts of forms and possibilities. Here are some interesting connections I have noticed, or would like to explore more: Computer based and synthesized or coded music, integration of electronics and micro-controllers like Lilypads in clothing and other textiles, and 3D design and fabrication in entrepreneurship programs.

   (Lee et al., 2020)

Computational thinking is not actually something that is new - it's part of other processes that we explore all the time in education. There will be many days in a year where algorithms, debugging, logical thinking will come up holistically in any number of subject areas. So, if it's already happening, why would we need to label it any further? I think this comes back to two things. 1 - Understanding that computational thinking is a process of thinking that we are doing all the time (even if by other names). But more importantly - 2 - we have to consider the end goal. In computational thinking we need to consider technology in our problem solving, we need to consider how our technological tools can help students, and that we can encourage our students to understand the technology that is available to us - to know how it works, to ensure that we can use it to look for better, more creative, more advanced, or more streamlined solutions.

Next episode, we have our first of three interviews. Kim Fick, a superintendent of education for the Sun West School Division sat down with me earlier this year. Kim was our principal until the end of last year, so I wanted to ask about her recent experiences working with other schools across the division.

Thanks for listening, I'm Stephen Hadden