K-12
BLUEPRINT

Bringing Education Up to Code with Computational Thinking: An Interview with Jane Krauss

Curriculum and Program Development Consultant Jane Krauss has come up with a tidy way of defining the term "computational thinking". As she explains it, computational thinking is posing a problem in such a way that a computer can help you solve it.

"It's a simple statement, but if you unpack it, everything you need is right there," she explains. "Jeanette Wing, a professor at Carnegie Mellon University, said that there were four pillars of computational thinking— decomposition to break down a problem, pattern recognition is another piece of it, abstraction and algorithms—and that they all add up to being able to pose a problem in such a way that you can solve it. We spend so much time on problem solving in education, but not as much time on problem posing and problem framing. Computational thinking is vital for everybody, and all kinds of minds should have a chance to dig into it."

> "Computational Thinking is posing a problem in such a way that a computer can help you solve it."

Teaching By the Book

Jane Krauss has been a fixture in the world of computer science education for years. By the time her and her team had the concept for the book *Computational Thinking (and Coding) for Every Student*, there wasn't another book like it. It was meant to be an entry-level invitation for educators and administrators to learn about computational thinking and what was possible in the classroom.

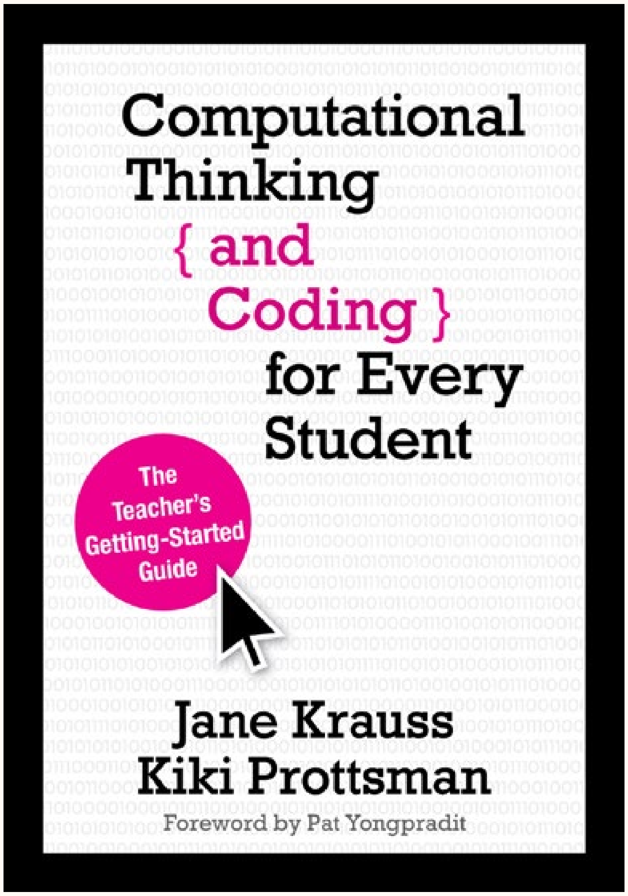
Krauss teamed up with computer science educator Kiki Prottsman to co-author the book. They felt that there was a groundswell of interest in computational thinking in general, yet—if you couldn't engage teachers—then what was the point? Krauss and Prottsman wanted to encourage teachers to try computational activities on their own—which is why there are so many included in the book—as well as pointing readers to opportunities beyond the book.

"Teachers, when you ask them to do one more thing, they just clutch at their throats," Krauss says. "They are very wary…is this the flavor of the week? Is this an entirely new discipline I'm expected to teach?"

Rising to the Challenge

One hurdle to professional learning is when a teacher doesn't feel confident even introducing a new concept to his or her students. Computational thinking, however, provides an opportunity for teachers to learn alongside their students. Block-based programming languages are another great entry point. MIT's Scratch, for one, is a visual programming tool that allows students to design games, model ideas, and conduct simulations. Educators can also take advantage of students who are already taking computer science in high school (and even middle school) and have them serve as a resource.

"Coding is the easiest way to get into computational thinking," Krauss says. "When a piece of code breaks, you have to figure out where you went wrong. The process helps students to persist and take feedback. Every kind of industry that a student will go into as a contributing participant in our society—every sector of our economy—is underpinned by technology now. And if students understand the underpinning of computing through coding, they will get better results when problem solving, as opposed to someone who is simply 'receiving' computer science."



Computational Thinking {and Coding} for Every Student

The Teacher's Getting-Started Guide

Jane Krauss
Kiki Prottsman

Foreword by Pat Yongpradit

Process Makes Perfect

Mathematician Seymour Papert—often considered the father of computer science education—viewed the computer as something to think and create with: a tool to produce something that's never been produced in the world before. With this in mind, Krauss considers computational thinking as another medium of expression and impact.

"If computational thinking concepts become a playground where students are able to drive their own interests to where they want them to go, they will be more successful," she says. "The computer should be something to think with and to express yourself with. If used this way, a computer becomes as fundamental as crayons and modeling clay. And everybody should have the opportunity to do that. It truly levels the playing field."

Finding Your Voice through Computing

Having worked at the National Center for Women in Information Technology, Krauss is committed to bringing traditionally underrepresented groups into computing. One initiative she sees as being particularly successful in broadening the reach of computational thinking is the Exploring Computer Science curriculum, developed using the research of Dr. Joanna Goode. The curriculum was implemented in the Los Angeles Unified School District with the intent of providing powerful computational thinking strategies in the district's most disadvantaged schools.

"Students find their voice within these activities, and end up programming their own apps," Krauss says. "It's an example of if you have equitable access paired with the right curriculum, then you will be more likely to see yourself in it. And tech companies are all over this. They know that when you have a diverse mix of race, age, gender and life experience in the production of services and products, you'll have better products because you're tapping into all of these different life experiences that people have: reflecting the different segments they represent."

One unexpected area where Krauss has seen a whole-hearted adoption of computational thinking approaches is Family and Consumer Science, formerly known as Home Economics. According to Krauss, the class had to make a case for how it is a modern and important subject, so curriculum creators tied most everything to STEM to show the relevance of consumer sciences.

"Take wearable computing, for example," Krauss says. "Home Ec was teaching sewing anyway, so Family and Consumer Science projects often include wearable Arduino microprocessors that can be used with LEDs and sound chips. There is also a push for mobile programming. MIT app inventor is a great tool since it's mobile. All kids have phones, even disadvantaged kids. So being able to program using their phones is wonderful."

Parting Thoughts

There is a bounty of resources for educators interested in teaching computer science principles. Educators can even check out computational thinking tools and resources from many public library maker spaces now, and a lot of tech companies are providing mini-grants for schools looking to build up a collection of programmable tools and devices over time. In fact, there are so many opportunities for bringing computational thinking into a curriculum, that some may find the prospect daunting. So what is Krauss' advice for teachers new to computational thinking who want to implement it in their classrooms?

"Give it a try yourself," she says. "There are so many teaching resources out there that you can't say it's an educational wasteland and that's why you can't do it! Give some of these activites a try. Try them with somebody on your team. Get a feel for what happens when you do it, and the satisfaction you feel from solving a problem using computational thinking."

"Give it a try yourself," she says. "There are so many teaching resources out there that you can't say it's an educational wasteland and that's why you can't do it! Give some of these activites a try. Try them with somebody on your team. Get a feel for what happens when you do it, and the satisfaction you feel from solving a problem using computational thinking."

—Jane Krauss, *co-author of Computational Thinking and Coding for Every Student: The Teacher's Getting-Started Guide.*