Pīkau Name: Computational Thinking: the International Perspective

Video Name: International definitions of Computational Thinking

Presenter: Professor Tim Bell

Organisations around the world have defined computational thinking in ways that we can apply them in schools. This definition here is from the Computer Science Teachers Association, which is an American organisation, and they call it an 'operational definition' – one that they can work with. They have got these six bullet points saying what the characteristics are of the process of computational thinking. They've said things like, 'It's formulating problems in a way that enables us to use a computer' and other tools to help solve those problems, that is the algorithm part. That's when we were looking at searching, it's 'How do I search a million things really quickly?' and then logically organising and analysing the data. When we were searching the phone book, it was much better that it was sorted into alphabetical order if that's how we were going to look things up.

Representing data through abstractions we saw in the sorting network, where students were comparing two values and working out which way to go. The 'abstraction' is that it could have been any two values, it could have been words it could have been numbers, and it could have been musical notes. Anything that can be compared, and we've 'abstracted' the general idea. Automating solutions, well that is where we actually get to program the thing. We actually get to make the computer do the work for us. Then identifying, analysing and implementing solutions. That's getting it onto the program and debugging the program and making sure that it works. Then generalizing and transferring the problem-solving process to a wide variety of problems, because when we teach specific examples in school, we know that those aren't the examples students are going to come up with. We want to develop a broader view of things for solving general problems.

The CSTA go on to say that there are particular skills that are associated with computational thinking, and they have listed these five, and you will see that they also relate to the front end of the New Zealand curriculum. It's about confidence dealing with complex situations. Being persistent when you've got a difficult problem, when you are working through it until you have actually solved it and not just having one go at it and hoping that it's the right answer. A tolerance for ambiguity – people will ask you to write programs, but they won't be very clear about what they really want. The ability to deal with open ended problems. 'Write a piece of software that helps people who are in hospital'; well that's a bit too open ended, so how do you narrow in and come up with something specific. The ability to communicate and work with others is so important because in the end you are designing digital systems for other people to use, and you need to understand what they need and how you can best help them by designing that system. Then of course digital systems are often so big that not one person can build them in a reasonable amount of time, so you are going to have to work with other people to work on different parts of the system; but to work on it in a way that when you bring them all back together they slot in nicely and you've produced a big solution that helps people.

Here's another take on computational thinking, this time from an organisation called 'Barefoot Computing', which is part of 'Computing at School', which is the UK teachers' organisation. What they have done is they've broken it up in a slightly different way but with the same kind of meaning. It talks about the concepts of computational thinking being: 'logic' – making good designs and plans and analysing things; then algorithms – algorithms always come up with computational thinking; decomposition – breaking things down into parts – that is quite an important skill where you take a big task and you break it into very small elements. For example we saw that with the sorting network, where we had a very small element (comparing two values, deciding which way to go), and that the big problem that was decomposed is 'How do we sort a whole lot of values into order?' We also saw it with the phone book where we decomposed the problem of searching millions of pages down to half as many, and half as many again. Patterns – looking for patterns, regularities that we can take advantage of to get the computer to do that

regular work for us. Abstraction – we've already talked about; and then evaluation, which is 'How fast is it?', 'How good is it?' and 'How easy is it for the person to use?'

Associated with that we have a set of approaches, which you can see also corresponds to the CSTA definition, but slightly different ideas here. One is tinkering – experimenting and trying stuff out; creating digital technologies is a very creative enterprise. Debugging – that's a really important one and we will spend quite a bit of time working at that, because when you design a new piece of software or a new piece of digital technology, you really want to find the bugs in it and track them down to get rid of them... anything that's not working properly, because in the end it's about delivering something that actually works, and that's a big part of it. Persevering – I really like that one because a lot of it is that you can't just build something and say 'I've done it'. You really need to evaluate it, you need to debug it, you need to check that it's working; and so perseverance is a very important quality that students will develop as they work on digital technologies. Then collaborating – again, everything is done with other people. You are designing things for other people, creating with other people, so a really important part of computational thinking.

Just to give one more international point of reference, let's have a look at a definition from Australia. This definition is a bit more convoluted, but it says that 'computational thinking describes the processes and approaches that we draw on when thinking about how a computer can help us to solve complex problems and create systems'. Again, it's all about taking someone's problems – something that they need done – and coming up with a digital system that helps to solve that problem for them. You will see that it talks about some of the ideas that we draw on to do that, and they will be familiar now because they match a lot of the ideas that we saw in the other international definitions; logical reasoning, algorithms, decomposition and so on. Hopefully you are getting the picture that this computational thinking thing is something of international interest. It is something that lots of people are working on, there has been lots of discussion about it at all sorts of levels around the world, and it means that there is also lots of resources that are being developed. As we work through the Pikau that we are offering you, we will show you some ideas that are really relevant to the New Zealand school environment, and some ideas – many ideas in fact – that are being developed locally that really situate computational thinking in a context of New Zealand, Aotearoa.