**Alicia Farrell:** Good afternoon, everyone, and welcome to today's session, Victorian Curriculum Mathematics F-10: Computational Thinking in Mathematics F-10. My name is Alicia Farrell and I'm the project officer in the Victorian Curriculum F-10 Unit here at the Victorian Curriculum and Assessment Authority. It is my great pleasure to introduce you to our presenter today, Dianna Chapman, who'll be leading our webinar today. But before we begin, we will start with an Acknowledgment of Country. I would like to acknowledge the traditional custodians of the many lands across Victoria on which each of us are living, learning and working from today. For myself and those of us in the Melbourne metropolitan area, we acknowledge the traditional custodians of the Kulin Nations. When acknowledging country, we recognise Aboriginal and Torres Strait Islander peoples' spiritual and cultural connection to country and acknowledge their continued care of the lands and waterways over generations while celebrating the continuation of a living culture that has a unique role in this region. I would like to pay my respects to Elders past, present and emerging, for they hold the memories, traditions, culture and hopes of all Aboriginal and Torres Strait Islander peoples across the nation and hope they'll walk with us on our journey.

So before we get into the presentation today, I'll briefly go over some housekeeping. Please note that the chat function is only being used to share relevant information and links from the VCAA. You will notice that a Q&A box has been set up. So please use these to put your questions and comments in, as this will help us ensure that all your queries are attended to and we do not miss any questions. When you use the Q&A box, please make sure you select "all presenters" so that all panellists can see your questions as they come in. And we will answer these questions in a couple of ways. Firstly, you may type a response directly into the Q&A box, which all participants will be able to view, or we will have a dedicated Q&A session at the end of the presentation where Dianna will address these queries.

The second part of our housekeeping is to let everyone know that this session is being recorded. A copy of both the recording and the PowerPoint, plus a transcript, will be loaded onto the VCAA's F-10 Resources web pages under the Professional Learning section. A copy of the recording and PowerPoint will also be emailed to participants in the coming days. So without further ado, Dianna, I'll hand over to you for today's presentation.

**Dianna Chapman:** Thank you, Alicia. Thanks very much, everybody, for attending this afternoon's session on computational thinking in the Victorian Curriculum, specifically in regards to Mathematics. Welcome, everybody. But first, I want to show you this poster which now looks quite blurry when it's been enlarged in a slideshow, so apologies. It doesn't look like that in in person. But this resource has been developed by the VCAA and it goes through the different areas of computational thinking. And it's a really great visual in terms of putting around your classroom. You can download it from the website. It has accompanying resources that I'm going to go through in today's presentation. But it just gives you a really good overview of what computational thinking is in regards to the Victorian Curriculum, in regards to Digital Technologies and Mathematics.

So, I was going to start off today with a little game, but it's going to be a little bit hard because I can't actually see, from my view, the chat function. So I'm just going to talk you through it. You can imagine actually doing this in your classroom and you can really do it at lots of different levels and age groups. So it is - guess the four-digit number. So, as a teacher, you would randomly pick a four-digit number and then you ask the class to guess the number, but the aim is that they have to do it with the least amount of questions possible. And the questions can only have yes or no answers. So they'd start off thinking or asking questions. Perhaps...some people would say, "OK, is the first digit an odd or even number?" And that breaks it down, you know, that sort of halves or breaks down the first digit, and then you can work out from there. Perhaps, "Is it divisible by two?" Or so on and so forth, different questions. Some may ask, "Is the whole number a...an odd or an even number?" The whole four digits.

So you can see that it starts to break down, and then from there, they will work out their guesses. They can do it a number of times, and you can see whether they get better and better at it, or...not better and better, but probably answer it with the least amount of questions each time. So it's a really good game to get classes...to get students thinking about how to break down a bigger problem, because getting a four-digit number just randomly is a really big problem. And to break it down into a smaller problem and then coming up with a rule at the end as well, is sort of furthering that idea and that game. So you can really extend students' thinking. So it's a really fun game. It has a lot to do with computational thinking, and that will make sense now. So, we were going to play it. But as I said, I can't see the chat screen from my view. So, yeah, I just want to start off with that as a bit of an introduction into getting you thinking today about computational thinking and how you can use different resources in your classroom. That's a really, really easy game to play.

So, what is computational thinking? So computational thinking is the thought processes involved in formulating a problem and expressing its solutions in such a way that a computer, either human or machine, can effectively carry it out. Now, I really, really like this definition because it uses the term "thought processes" for formulating the problem and expressing a solution. And so that's what you do in your head. OK, so computers and programming is a way to make a solution concrete so that it can be run on a computer or a machine. But computational thinking is what happens first. So the thinking's first and foremost, and that's what we as humans do. So that's why I really like this definition, because it's about the thought processes, and often it gets confused with programming, and do we need to have a certain software to be able to do computational thinking? Actually, the processes that happen before it goes into programming and being used in that setting - so it's those thought processes, it's the thinking. And how do you systematically and rigorously solve problems? So it can be used across all different curriculum areas. So it's not just Mathematics, which is really nice tool to have. But obviously in mathematics we would use it as an approach to problem-solving. So you can take computational thinking with or without the use of technology as well, and that's where that idea of it being thought processes involved in formulating the problem is really, really nice. So that's a... I really like that definition.

So, with computational thinking, there's four techniques that can be used. We have decompositions, pattern recognition, obstruction and algorithms. So, decomposition, that's that breaking down of a complex problem into smaller and more manageable parts. So it's breaking it apart - those Venn diagrams, those... If you've got that really big problem, like that four-digit game, so you've got the four digits that you have to guess, but to break it down into a smaller problem, we might look at the first digit and see what I come up with there, or the last digit to see what I come up with there. And so then it's a more manageable problem. So, then, by breaking it down, or decomposing, the problem is more manageable. And then from there, you can start to think about patterns that might appear. So it's looking for similarities among the different subproblems that you might have. And then the abstraction. So that's focusing on the important details and getting rid of the noise, I guess, around the problems. So you're only concentrating on the information that you really, really need to solve the problems. And then algorithms. So, a step-by-step solution. It's developing that step-by-step solution, a rule that you might follow to be able to solve that problem.

So there are the four different parts, and I'm going to go into them in more detail. So that's just a really thick overview of those four... I guess they're the cornerstones of what computational thinking is. So, a complex problem at first glance, just sort of a problem that we don't really know how to solve easily. So computational thinking involves taking...taking that problem and breaking it down into a series of smaller problems, so that's the decomposition. And each of these smaller problems can then be looked at individually. Considering how similar the problems are, then you're looking at sort of patterns within the problems that you have, that pattern recognition. And then you're focusing on the important details, so that's the abstraction. And then the simple steps or the rules that you're going to follow, so that's the algorithm.

So, this is really interesting in that...computers obviously don't think for themselves, so we have to work out what we're going to put into a problem before it is solved, and that's what computational thinking is. So, sort of simply put, I guess you'd say that programming tells the computer what to do and how to do it, and computational thinking enables you to work out actually what you want to tell the computer to do, and computers being not necessarily an electronic machine, but also the brain and the way that the brain can function as well, in that you're breaking it down into smaller things, step-by-step rules that you're going to follow. So that's where you don't necessarily need technology for computational thinking. So, for example, working out what you need to do... An everyday example is that you might be meeting a group of friends somewhere that you've never been before. So you'd have to think about how you're going to get there, what transport you're going to take. Are you going to drive? Are you going to walk? What's the quickest way? Say you need to pick up something on your way from the shops. Where are the shops? Are they close to where I'm going? So all these steps along the way is building up that idea and breaking down that problem into smaller little problems to be able to solve. You can work out which way you're going to go - the shortest, the quickest, if you're going to go past the shops and all that type of thing.

So it's the planning. It's really about the planning part of a problem, is computational thinking, But I really wanted to get that across, because I know there's often a lot of confusion about what it actually is. So hopefully by giving everyday examples like that, that planning phase... The planning phase towards that is the computational thinking, and then the actual doing, so when you're actually following the directions that you're going to take to meet your mates is like the programming part of it. And it's a really important and integral part of problem-solving, too - to break it down and to plan out how you're going to solve a problem.

Alright, so I'm going to have a look at the...at the four areas here in a bit more detail. And I'll give you some examples of how they can work. So... Or how you could visualise them as well. So decomposition, it's something that we probably inherently do in our day-to-day lives without maybe even realising it. So the concept probably already exists within our students. So it's important for them to be able to learn and recognise that that's what they're doing in that certain situation and be able to then use it, and so not being so overwhelmed when they are confronted with a problem to solve that might seem too much or too hard or too difficult. So that idea that... OK, well, you know, perhaps if I'm planning a meal at home and I'm inviting some friends over - a barbecue, say... Well, this example is probably... At the moment, we're not allowed people over at the moment. But if we were allowed to have a barbecue and have people over, you'd have to think about what you wanted to cook. You'd have to think about if you wanted to ask people to bring certain things, make sure that you don't double up if someone's going to bring dessert, but someone's bringing salad. You don't want five salads. I'd like five desserts, but not five salads. So it's that...that process of really breaking down that problem. So, you can see there the example - if there's a problem, you can break it down into two problems, so 1 and 2. And then from problem 1, you can have one 1A or 1.1, and 1.2... Breaking it down into those smaller, more manageable areas to concentrate on and to think about.

So the pigeonhole principle is another idea around decomposition as well. So that's about counting, but it means that there's more... Well, more pigeons than holes. So you're going to have to put them...double up pigeons in a hole. So it's...illustrating how you can...how you can actually break down that problem if there are more. Would you leave them out, or you're going to put them in the box or not in the box? So, again, another example of how we can use decomposition. So, another area in maths that you could use it in is if students were asked to find the area of a shape, and they could, if it was an irregular shape, they could have a look at how to use an area of a known shape to put into that and use that known area to be able to put it into the larger shape... ...to where it... (SOUND BREAKS UP) ..be...area of the irregular shape. Or like the example that we had at the beginning, the four-digit number. So breaking it down, working out each one. Really good for place value, obviously, as well. That's also when you can sort of say to students that you're recognising place value. Say you had a number - 456. By breaking it down into hundreds and units, you would know...that that is a form of decomposition. But that just gives you a couple of examples there where you can use it in the classroom, or where it is used in the classroom, or you might not know.

So it's really important, then, to start to recognise that that is actually decomposition. And what you're doing there is decomposition, and that's part of problem-solving, and part of growing those problem-solving skills that we need. So we want to also then recognise patterns when we're problem-solving, because this will help with recognising what the problem is that we need to solve. So by classifying data, or organising data, that data representation and interpretation is an example of pattern recognition. So you'd recognise patterns and connections with different pieces and you'd be able to then create, say, data. So that one on screen now is the water storage. There's also obviously lots and lots of other ways of representing data. So... And the analysis of that and what that actually means is part of pattern recognition. So, again, it's breaking it down, computational thinking. You may not realise that your students are actually doing it, but they are. And it's getting them to recognise that when they're doing these types of work, that that's actually what they're doing and what they're being involved in while they're doing that, and how important that is to recognise that, so to be able to... (SOUND BREAKS UP) ..pattern recognition. And why do we need to know about pattern recognition? And that's sort of building up that concept for them.

And another form of pattern recognition and an example that you could use in class is when, if you're making paper chains, and you're going to repeat a pattern. So say you wanted a red, blue, green, red, blue, green. That's pattern recognition, and that's starting from those younger year-levels in maths, so recognising that pattern, when to re-use the next colour change. That leads on to things like number sequences, so there's pattern recognition in number sequences - two, four, six, eight - what comes next? And build up the difficulty, obviously, from there in your number patterns. So development in your number sequences and patterns. So there's...lots of examples that have pattern recognition is used in maths. But I guess the important thing is to recognise that you are actually using it and for students to understand what they're actually doing when they're doing it.

This is probably one of my...favourite ones that I relate to - getting rid of all the noise. (LAUGHS) Just focusing on the essential problems is abstraction. So it's really about removing that non-essential information. Succinctly, we're drawing a diagram here, like the example there is - draw a diagram of your home network. So what do you need? There's basically the wires and things like that that are there that you do need. But that's not going to be part of my drawing. I'm just going to show you the really important facts about what I need to be able to set up a home network. And that's that abstraction. I really like the train map as a way of describing what abstraction is as well, because if I'm a passenger on a train...I only really need to know where my next...where I get off, or maybe what the stop before is so I can keep an eye out for it. So the details in these maps - it's not detailed. You don't have the terrain, you don't have the distance between each stop. So it's getting rid of that information and only giving the passengers that really...that information that they need to able to solve that problem of, "When am I going to get off?" and to be able to plan your journey. So I think that train one is a really good example of abstraction. The other details around - distance, you know, if it's the tunnel, the depth, you know, you don't know... You don't need to know that as a passenger. It might be really interesting information to have, but as a passenger I don't really need to know that if I'm planning my journey. So that's around... You can probably think of lots of other examples of using abstraction...as well.

Um, you then have algorithms, so that's setting up those rules to follow. When am I going to use the rules? Obviously, I've worked out that there's a pattern. I've worked out that this is the way I'm going to do it. How am I going to describe that, moving on? I'm going to write a rule. It could be in a flow chart. So, for example, there we've got, you know, the start, the read user, the selection... If this happens, yes. If this happens, go over there. So flow charts are a really good way of showing...of an example of showing algorithms as well. Um, and... Obviously, that division as a repeated subtraction... I don't why that N's cut off there. Anyway... So, here you can see that...the multiplication of the positive integers can be considered as repeated additions. And in a similar way, division of a positive integer by a smaller positive integer can be considered as repeated subtraction. And so then you're coming up with that rule that you're going to follow - which is all on that example...there. And I'm hoping that my...internet works on my next slide, 'cause I'm going to show you some resources from the Victorian Curriculum...of where we have some fantastic resources available. And I'm hoping... Maybe... I'm hoping that you can see this screen with the computational thinking...which...

**Alicia Farrell:** Yes, we can see it, Dianna.

**Dianna Chapman:** Can you see it now? Thank you. Out into the ether, I send you!

**Alicia Farrell:** No problems.

**Dianna Chapman:** It's so strange not hearing a response. It's a very strange way to present. Um, so this is where the poster is that we...that I showed you earlier. So here you can obviously get the different sizes and things like that. But this is the exciting part for you, in that the VCAA has created resources that go through the mathematics content descriptions which are related to computational thinking. And it shows you...links to the achievement standards for that particular content description and how...and then learning activities related to it as well. So I'm going to...jump into one. I'll show an example for a primary and then an example for a secondary. So I'm just going to go mid-level here - we'll go to Level 4. It might take a while to load up - apologies. I need to zip across so you can see here. So, Alicia, can you see that now, that document?

**Alicia Farrell:** I can, yes.

**Dianna Chapman:** Awesome, thank you. So, here you can see the resource for Level 4. So, you could see earlier that we've got them for all the levels. We've gone through all the content descriptions that relate to computational thinking. So this one here...is... The content description here is define a simple class of problems and solve them using an effective algorithm that involves a short sequence of steps and decisions. And when we think about that, you can think, obviously, that that whole idea of algorithms - what's the rule that I'm going to follow? And so setting...setting up students to understand that we're going to break down that problem into a smaller problem and create a rule that we're going to follow.

So here the example of the lesson is - more money, more problems. And so it's about sorting coins. So, again, it's that pattern recognition, it's about breaking down the problem into smaller problems. And again, you can build on these resources as well - like you might see this resource and think, "Oh, that'd be great for my class, but my other class might do better with, maybe, shapes," if you were doing it from smallest to largest, or things like that. So this is one of those sorting activities. So, with two coins, maybe you'd sort them by size, or you could start sorting them by value. And then you move on, building on that, and sort three coins. And there's... In these resources, there's discussion points that you may have for the questions that you could raise with your students as well. So you can see, there's a lot of detail in these resources, and it goes through...really, the underpinning ideas behind computational thinking, and it's also recognising which content descriptions are related to computational thinking, so you can then apply your knowledge of what that is for you and how you can apply that in the classroom. So, really, really useful...tools for...for teaching.

So, I'm going to close that one. And we'll do a Level 9 now. So they're... I think they're a really good starting point to get you thinking about...how we can use computational thinking in the classroom and how... You're probably looking at this saying, "I probably already do this," or in some way you are already doing it. And it's now then, if you sort of have recognised that, it's then saying, "OK, then, how can I get my students to really engage in that problem-solving technique of computational thinking?" So this one is applied to structures to solve real-world problems. So that's the content description. For this one - students will use sorting algorithms to facilitate finding the median of a set of numbers. So, again, you can do this with or without technology. So here it's suggesting that you could use Python or Wolfram, if you have that in your classroom. So... Again, just giving those resources. It has great little YouTube trips... Clips, sorry. And, yeah, again, those discussion points around...around it.

So I think the idea of computational thinking...is really probably...is....part of what we do already, so it's just understanding and getting that knowledge of what it actually is and how you can actually then use it in the classroom, or if you're already using it in the classroom, how to really hone in on those skills and build that skill set to be able to...go for each of the four parts of computational thinking. This clip is a really nice example of...sorting algorithms. So I'll show you... Hopefully this works, too. And it's something, again, that you could do easily in your classroom with your students, and it's a really great visual way to explain, in this case...sorting. So they go through the different types of sorting. You can see there's 14 people. We can see that they're trying to work out, "OK, we want to get them from...largest...smallest to largest." And the different techniques that you could use to get our numbers that have been mixed up. So, in this one, they're looking as they go along...and inserting where it should go, compared to the other digits. So a really good activity that you could use in the classroom. Glad I started speeding it up. I wonder if I can go to the next one. Almost there. And again, this game goes back to that sort of beginning game that we were going to play as well, the four digits, guessing the four digits, the least amount of guesses that you might need if you were going to play that game.

This is another way, another technique...that the sorting wizard is going to do. And you could get students to...work this out themselves. You know, they might come up with these different techniques, and then just discussing what those different techniques are and what they're called, breaking down that problem of, "OK, well, we've got 14 people and we need to get them from smallest to largest. What... What's the quickest way?" I might just forward this. And... Yay! They've done it that time. So, again, so that rule, it's that algorithm. Looking at your neighbour... Who's smaller, who's larger? Grouping them. And we get through and through. So, you can see... I really like that, because it is a visual way of different ways of coming up with a rule so that... It goes through all sorts of different rules that you can use, or algorithms that you can use. So it's a really nice activity that you can use in the classroom as well.

And there's so many different content descriptions that that could apply itself to, or lend itself to, in that pattern recognition, in that breaking down the problem into smaller problems, in the getting rid of the information that we don't need, that abstraction, and then the making the rule at the end, which... Sometimes the rule at the end, it can get a bit more complex later...a bit later on. But I think if...from... If we're starting with the foundations of what computational thinking are and we're introducing students to those different parts of it early on in the curriculum, which...which we do, and if we start to get them to be able to recognise that that's what they're doing, then when it comes to the more complex maths later on, they're going to have that recall of, "Oh, yeah, that's right. I know that that's a type of abstraction." Or they'll start to use that language, which means they're really recognising the skill that they have to be able to do...and to build on that skill. So that's what's really important, I think, about computational thinking.

So I've gone over each of the four components today and I've shown you where to find the resources that are available, and I'd really encourage you to have a look through those resources in detail. It's hard to see through a presentation like this. So I really do encourage you to have a look at those links yourself. And again, it's about recognising for you that you are probably doing a lot of this in your class already, or if you're not, then you can introduce activities like that, or it's about understanding that, you know, you don't necessarily have to use technology to be able to...to have those resources for computational thinking. It can be done. And the idea that it is about that planning, so it's before, you know, before you're telling them what to do. So it's that creation of the rule is basically what the outcome of computational thinking is. When you've gone through all the steps, you come out with that rule at the end, and then that's...that's the programming part and that's what you need...is used for programming.

So it's the thinking, and it goes back to that... I'll go back to it now. The definition at the start around...computational thinking in that it's those thought processes, which I think is really, really important. And that definition sort of rings true for me and I think is a really nice way to explain what computational thinking actually is. So that comes to the end of the presentation today. As I said, there's a lot... those resources to go through which are really, really useful. And I hope that you have got a lot out of today and you've got something that you can take into the classroom as well, which is what we always would like to be able to do here in these sessions as well. So thank you, everybody, for attending.

[Copyright Victorian Curriculum and Assessment Authority](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx) 2020