Computational and
algorithmic thinking
in Mathematics

Unpacking the content descriptions

Level 10A



Authorised and published by the Victorian Curriculum and Assessment Authority
Level 7, 2 Lonsdale Street
Melbourne VIC 3000

© Victorian Curriculum and Assessment Authority 2019

No part of this publication may be reproduced except as specified under the *Copyright Act 1968* or by permission from the VCAA. Excepting third-party elements, schools may use this resource in accordance with the [VCAA educational allowance](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx#schools). For more information go to: <https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx>.

The VCAA provides the only official, up-to-date versions of VCAA publications. Details of updates can be found on the VCAA website: [www.vcaa.vic.edu.au](http://www.vcaa.vic.edu.au).

This publication may contain copyright material belonging to a third party. Every effort has been made to contact all copyright owners. If you believe that material in this publication is an infringement of your copyright, please email the Copyright Officer: vcaa.copyright@edumail.vic.gov.au

Copyright in materials appearing at any sites linked to this document rests with the copyright owner/s of those materials, subject to the Copyright Act. The VCAA recommends you refer to copyright statements at linked sites before using such materials.

The VCAA logo is a registered trademark of the Victorian Curriculum and Assessment Authority.

Contents

[Explicitly teaching computational and algorithmic thinking 4](#_Toc16695223)

[Overview of the resource 5](#_Toc16695224)

[What’s behind the door? 7](#_Toc16695225)

[Appendix 1 12](#_Toc16695226)

[Appendix 2 13](#_Toc16695227)

[Appendix 3 14](#_Toc16695228)

Explicitly teaching computational and
algorithmic thinking

The *Computational and algorithmic thinking – Unpacking the content descriptions* resources unpack the Victorian Curriculum F–10 Mathematics content descriptions that address computational thinking and algorithms at each level in the Patterns and algebra sub-strand of the Number and Algebra strand.

Each resource provides teachers with links between one Mathematics content description and extract from the achievement standard related to computational and algorithmic thinking and a teaching and learning activity that is designed to develop computational thinking and problem-solving skills and produce corresponding algorithms in a mathematical context. Teachers can also find excerpts from the Victorian Curriculum Mathematics and Digital Technologies glossaries in [Appendix 3](#Appendix3).

The resources have been developed with respect to teaching in the Mathematics learning area of the Victorian Curriculum and they also include suggestions how these activities could be extended to the Critical and Creating Thinking and Digital Technologies curriculums.

Teachers will find detailed ideas about how to integrate Mathematics with one or both of Digital Technologies and Critical and Creating Thinking in [Appendix 1](#Appendix1) and [Appendix 2](#Appendix2) respectively.

Overview of the resource

**Curriculum area and level:** Mathematics, Level 10A

**Strand and sub-strand:** Number and Algebra, Patterns and algebra

**Content description:** Devise and use algorithms and simulations to solve mathematical problems ([VCMNA358](https://victoriancurriculum.vcaa.vic.edu.au/mathematics/curriculum/f-10#level=10A&search=b0151a0a-375f-4f5d-a236-a4c101092663))

**Achievement standard (extract):** [Students solve problems numerically, graphically and algebraically] … to model situations and solve practical problems. (Level 10 achievement standard)

**Title:** What’s behind the door?

**Timing:** 2 or 3 lessons (approx. 100–150 minutes)

**Description:** Students research mathematical problems and create simulations to explore these problems.

 In these activities, students devise their algorithms and carry out their simulations by hand, using some combination of:

pseudo-code

flowcharts

chance experiments (such as with cards or dice).

 Students first need to plan how to represent the mathematical problem, then they create a simulation that can be used to collect data on the problem. Finally, students analyse their data and propose a solution to their mathematical problem.

**Learning objectives:** Students can:

create an algorithm to explore a mathematical problem

devise a simulation to test a hypothesis about how the mathematical problem works

analyse their collected data to come to a conclusion as to the most likely or most well supported solution for the mathematical problem.

**Resources:**

paper, pens (for planning visually)

dice, cards, coins

computer or other ICT (e.g. mobile phone, iPad) for simulating random events and/or collecting and analysing data (optional)

if using a programming language, access to a computer with internet connectivity and the required software/coding platform (optional)

Some students with experience using a computer algebra system or a programming language may wish to complete this activity using digital technology. Encourage students to select a computer language, application or platform that they are comfortable with, and get them to modify an existing simulation or create one themselves. For example, students could use Microsoft Excel, Scratch (block code and animation), Mathematica or Python.

What’s behind the door?

Scaffold and support

Teachers are encouraged to use and clearly define the terms ‘simulation’ and ‘algorithm’ in this activity to reinforce the process of following a set or sequence of instructions to solve a problem (see the [Victorian Curriculum Mathematics Glossary](http://victoriancurriculum.vcaa.vic.edu.au/LearningArea/LoadFile?learningArea=mathematics&subject=mathematics&name=Mathematics%20Glossary.docx&storage=Glossary) or [Appendix 3](#Appendix3)). Teachers could choose to define these terms themselves for the class or give students the opportunity to come up with their own agreed class definitions before, during or after the activity and compare them with the definitions in the glossary.

Some additional vocabulary has been included below.

**Chance and likelihood**

The relative frequency of an event is the chance or likelihood of the event occurring. This may be expressed qualitatively, using terms such as ‘impossible’, ‘no chance’, ‘not likely’, ‘an even chance’, ‘odds-on’, ‘likely’ or ‘a certainty’. Relative frequencies may also be expressed quantitatively, using numbers on a scale from 0 (impossible) to 1 (certain). These numerical values are often expressed as fractions (such as ½), ratios (such as 2:3), decimals (such as 0.87) or percentages (such as 40%).

**Simulation**

Simulation is the process of modelling an event using a device or technology. For example, if on past performance two players are equally likely to win a game of tennis, then a sequence of games between the two players could be simulated by successively tossing a fair coin (for example, heads represents player A wins, tails represents player B wins) or randomly selecting numbers from the list of natural numbers and noting whether the result is even (player A wins) or odd (player B wins). This could be represented using a tree diagram.

**Probability**

The probability of an event is a number between 0 and 1, inclusive, that indicates the chance of something happening. For example, the probability that the sun will come up tomorrow is 1, the probability that a fair coin will come up ‘heads’ when tossed is 0.5, and the probability of someone being physically present in Adelaide and Brisbane at exactly the same time is zero.

**Problem-posing and problem-solving**

This is a two-part process: first, formulating a problem in such a way that it is amenable to mathematical analysis; second, applying mathematical reasoning to the development of a solution (or solutions) to a given problem.

Activity 1

Describe the following problem to students. You could demonstrate the problem by acting as a host on your own ‘game show’ with students as contestants.

There are three closed doors on the game show and behind each door there is a prize. The contestant can’t see the prizes, but the host knows where each prize is. Behind one of these three doors is a car. Behind the other two doors are goats.

The contestant is asked to select a door to win the prize behind it. They don’t want to win goats! They choose a door, but the host doesn’t open it.

The host knows where all the prizes are. From the other two doors, the host opens a door that has a goat behind it.

There are now two closed doors – the one the contestant chose and the door that the host didn’t open.

The host asks the contestant: ‘Do you want to switch your original choice of door?’

**Discussion prompts**

Get students to discuss in small groups what they would do in this situation.

* Imagine you are a contestant. What would you do?
* Would it make any difference if they switch?
* Should they switch? Why?
* Should they stay with their original choice? Why?

Now, the contestant makes their choice and opens the door. Did they win?

You could repeat this scenario a number of times and encourage students to keep thinking about the outcomes.

**Discussion prompts**

After repeating the scenario a few times, get students to discuss in small groups what they would do in this situation.

* After seeing this problem simulated a few times, have you changed you opinion about what they should do? Why or why not?
* What questions do you now have about this problem?

After encouraging students to consider these discussion prompts, have students watch an explanation video on the Monty Hall problem, such as [‘Monty Hall Problem – Numberphile’](https://www.youtube.com/watch?v=4Lb-6rxZxx0).

**Discussion prompts**

Get students to discuss the following in small groups.

* What was surprising about this solution?
* What do you think the term ‘counter-intuitive’ means? Why could we say this is a counter-intuitive problem?

► **Cross-curricular links**

See [Appendix 1](#Appendix1) and [Appendix 2](#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Activity 2

In this activity, students will work in pairs or small groups to do three things:

1. Devise a **simulation** to test a problem.
2. **Collect data** from the simulation.
3. **Propose a solution** to the problem by analysing and reflecting on their data.

There are many ‘counter-intuitive’ problems in Mathematics that students could consider for this activity, aside from the Monty Hall problem. Below are links to resources providing background or explanations on a few of these problems:

* [The Cereal Box Problem](https://mste.illinois.edu/reese/cereal/cereal.php) (Office for Mathematics, Science, and Technology Education, College of Education, University of Illinois webpage)
* [The Birthday Problem](https://mste.illinois.edu/activity/birthday/) (Office for Mathematics, Science, and Technology Education, College of Education, University of Illinois webpage)
* [Dice duels (many variations)](http://archive.stats.govt.nz/tools_and_services/schools_corner/Activities/Interactive-Games.aspx#si-6) (Stats NZ webpage)
* [Snake Eyes Riddle](https://www.youtube.com/watch?v=lTJydnfGyyk) (MindYourDecisions, YouTube video)
* [The 3 Cards Riddle](https://www.youtube.com/watch?v=1IFM1ngwEb0) (MindYourDecisions, YouTube video)

How students create their simulation is entirely up to them. The may wish to use any kind of manipulatives, such as:

* playing cards
* dice
* coins
* dominoes or numbered cards
* props (such as the doors in the Monty Hall problem)
* technology (such as Microsoft Excel)
* online simulations of random numbers, such as those at [random.org](https://www.random.org/)

Encourage students to:

* be creative – think outside the box!
* research ideas on simulating their problem – they might find tutorials online such as this one for the [Monty Hall problem using dominoes](https://www.youtube.com/watch?v=A4wCuVVI5aQ)
* reflect on their simulation – does this seem reasonable?
* collaborate – share ideas with each other and be constructive in their criticism.

Once students have come up with their simulation, they should run it – over and over! More data is better. Remind them to accurately record their findings.

When the students have run a number of simulations, they should analyse the data they have collected.

Students should then present their findings. They might like to do this by demonstrating their simulation to the class. As they explain their simulation, they should:

* explain why it works to test their problem
* discuss what their findings indicate
* provide the solution they have reached based on their findings
* suggest future directions for this investigation.

**Discussion prompts**

Get students to discuss the following in pairs of small groups.

* What was surprising about this solution? Was it what you expected?
* Why could we say your chosen problem is a counter-intuitive problem?

Encourage students to compare and contrast their problems, simulations and solution strategies with those of other pairs or small groups.

Challenge and extend

**Simulations and programming**

Watch online tutorials that show programming for simulations . For example:

* ['Dice rolling simulation using Excel'](https://www.youtube.com/watch?v=KjRj0Q8SIMI) (rodcastmath YouTube video)
* [using Scratch to animate and simulate the Monty Hall problem](https://scratch.mit.edu/projects/10366297/#editor) (MIT Media Lab) – click on ‘see inside’ to see the code used in this project.

**Probability theory**

For an alternative extension, have students research and explain the probability theory that underpins their mathematical problem, to support (or challenge) their findings from their simulations. There are a range of explanations of probability theory online. For example, watch the [Monty Hall Problem video at this link](https://brilliant.org/wiki/monty-hall-problem/).

Encourage students to link their research to their past experiences with probability and chance.

► **Cross-curricular links**

See [Appendix 1](#Appendix1) and [Appendix 2](#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Reflection

Students share their findings from their research with others, compare their main findings and reflect on these findings.

With the class, the teacher could ask the students to discuss the difficulties they encountered and discuss different strategies that students used to overcome these.

* What were some difficulties you faced in this task?
* How did you overcome these? (What were some strategies you used?)
* What were some skills you learnt that helped in a similar situation?
* Can you think of other problems where a simulation might be used to help solve a complicated problem?

With the class, in small groups or individually, the teacher should ask students to reflect on their algorithmic thinking. Ask students:

* How have you demonstrated algorithmic thinking during this activity?
* How does your simulation work to answer your chosen mathematical problem?
* What does the simulation do well? Why?
* What doesn’t the simulation do well? Why?
* What do you think was the purpose of this activity?

**Tip:** Reflection is a very important part of any computational thinking–focused activity because it encourages students to consider the different aspects of the task, such as defining the problem, selection of tools and processes, problem-solving, teamwork and verifying their solution. This helps students reflect on the process of their own learning (meta-learning) and how the skills and tools they have used might be transferred to other contexts.

Additional teaching resources

* The unit [Understanding Behaviour through Simulation](https://www.resolve.edu.au/algorithmic-thinking-simulation?lesson=1854) on reSolve is part of the special topic [Mathematics and Algorithmic Thinking](https://www.resolve.edu.au/node/1851). It demonstrates simulation as a complement to traditional mathematical methods to understand real-world scenarios that seem to be beyond our mathematical grasp. The emphasis is on fundamental questions of behaviour in biology and sociology, where simulation has enabled significant scientific discoveries. Topics include human segregation, making good choices given limited data, game theory, animal behaviour and evolution of social traits.
* [PHET Interactive Simulations](https://phet.colorado.edu/en/simulations/category/math) (University of Colorado, Boulder) contains a large number of interactive simulations for mathematical problems, including Plinko, projectile motion and proportions. Note, some simulations require Java.
* StudyIt (New Zealand Ministry of Education) is an online resource covering New Zealand curriculum. [This webpage](https://studyit.govt.nz/Maths/level/5/standard/2.13/subjectContent) is a great resource for investigating a situation involving elements of chance using a simulation. The resource provides links to a number of interactive examples of simulation including the Cereal Box problem, Queues, the Birthday problem and the Monty Hall problem.

Appendix 1

Suggestions for explicitly teaching Digital Technologies (stimulus only)

**Curriculum area:** Digital Technologies

**Strand:** Creating Digital Solutions

**Band:** Levels 9 and 10

**Content description:** Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases [(VCDTCD052)](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCDTCD052)

**Achievement standard (extract):** They design … algorithms … and test modular programs.

**Suggestions that link to the activities:**

* Creating algorithms that use various functions and data structures.
* Testing the expected output of algorithms using trace tables and desk checking if necessary in order to make modifications and record results.

See also [Unpacking Digital Technologies Content Descriptions](https://www.vcaa.vic.edu.au/Documents/viccurric/digitech/Unpacking_the_Content_Descriptions/Unpacking_Digital_Technologies_Content_Descriptions-9-10.docx) for Levels 9 and 10.

Appendix 2

Suggestions for explicitly teaching Critical and Creative Thinking (stimulus only)

**Curriculum area:** Critical and Creative Thinking

**Strand:** Questions and Possibilities

**Band:** Levels 9 and 10

**Content description:** Suspend judgements to allow new possibilities to emerge and investigate how this can broaden ideas and solutions ([VCCCTQ044](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTQ044))

**Achievement standard (extract):** They demonstrate a willingness to shift their perspective when generating ideas, resulting in new ways of perceiving solutions.

**Suggestion that links to the activities:**

* Being open to the results of the simulation, supporting alternative solutions to what may have been expected, particularly when developing simulations for counter-intuitive problems in probability such as the Monty Hall problem. Did this ‘openness’ result in a useful broadening of ideas and way of perceiving solutions?

Appendix 3

Excerpts from the Victorian Curriculum Mathematics and Digital Technologies glossaries

There are some commonalities in the terms used when explicitly teaching computational and algorithmic thinking in Mathematics and Digital Technologies; however, there are also some subtle but important differences in the definitions of terms. Some of these common terms and their definitions are listed below, under the two different curriculum areas.

If you are going to explicitly teach Mathematics, please refer to the [Victorian Curriculum Mathematics Glossary](http://victoriancurriculum.vcaa.vic.edu.au/LearningArea/LoadFile?learningArea=mathematics&subject=mathematics&name=Mathematics%20Glossary.docx&storage=Glossary). If you are also going to explicitly teach Digital Technologies, refer to the [Victorian Curriculum Digital Technologies Glossary](https://victoriancurriculum.vcaa.vic.edu.au/LearningArea/LoadFile?learningArea=technologies&subject=digital-technologies&name=Digital%20Technologies%20Glossary.docx&storage=Glossary).

Mathematics

Algorithm

An **algorithm** is a process that can be carried out mechanically, using a well-defined set of instructions, to perform a particular task or solve a type of problem. Examples of mathematical algorithms include processes for tasks such as ordering a set of numbers from smallest to largest, multiplying many-digit decimal numbers, factorising linear expressions, determining which of two fractions is larger, bisecting an angle, or calculating the mean of a set of numbers.

Algorithmic thinking

**Algorithmic thinking** is the type of thinking required to design, test and evaluate problem-solving processes in a systematic way, using algorithms.

Coding

A process by which algorithms are represented for implementation. For computers, this is done using a coding language such as block coding, C++, JavaScript, Python, Wolfram Language.

Computational thinking

In this context, computational thinking is considered to be linked to algorithmic thinking. This type of thinking is usually considered specific to computers which involves solving problems, designing systems and implementation.

Sequence (number)

A **sequence** is an ordered set of elements such as numbers, instructions or objects. From an algorithmic point of view, a sequence is an ordered set of instructions or actions.

Unplugged

A commonly used term for computational thinking activities carried out without digital technology. “Unplugged” representations of algorithms may include structured mathematical processes, English representations (steps) or flowcharts.

Digital Technologies

Algorithm

A description of the steps and decisions required to solve a problem. For example, to find the largest number in a list of positive numbers:

1. Note the first number as the largest
2. Look through the remaining numbers, in turn, and if a number is larger than the number found in 1, note it as the largest.
3. Repeat this process until complete. The last noted number is the largest in the list.

Flowcharts are often useful in visualising an algorithm.

Computational thinking

A problem-solving method that involves various techniques and strategies in order to solve problems that can be implemented by digital systems, such as organising data logically, breaking down problems into components, and the design and use of algorithms, patterns and models.

Structured English

The use of the English language to describe the steps of an algorithmin clear, unambiguous statements that can be read from start to finish. The use of keywords, such as START, END, IF, UNTIL, provides a syntax similar to that of a programming language to assist with identifying logical steps necessary to properly describe the algorithm.

An example of the use of structured language can be demonstrated using the following problem:

Description of the problem:

Describing the decision a person makes about how to get to a destination based on the weather and the distance from their current location to their destination.

Structured English example:

START

IF it is raining outside THEN

 Catch the bus

ELSE

 IF it is less than 2km to the destination THEN

 Walk

 ELSE IF it is less than 10km to the destination THEN

 Ride a bicycle

 ELSE

 Catch the bus

 ENDIF

ENDIF

END

The Structured English description can easily be translated into code using a programming language and accurately captures the logical elements that must be followed to answer the question posed.