Computational and   
algorithmic thinking   
in Mathematics

Unpacking the content descriptions

Level 8



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](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTQ003#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](https://www.mathplayground.com/grade_1_games.html).

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](mailto: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](#_Toc23330669)

[Overview of the resource 5](#_Toc23330670)

[Divide and conquer 6](#_Toc23330671)

[Additional teaching resources 13](#_Toc23330672)

[Appendix 1 14](#_Toc23330673)

[Appendix 2 15](#_Toc23330674)

[Appendix 3 17](#_Toc23330675)

[Appendix 4 19](#_Toc23330676)

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 8

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

**Content description:** Use algorithms and related testing procedures to identify and correct errors [(VCMNA282)](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCMNA282)

**Achievement standard (extract):** Students use efficient mental and written strategies to … carry out the four operations with integers …

**Title:** Divide and conquer

**Timing:** 1 or 2 lessons (approx. 50–100 minutes)

**Description:** Students explore the divisibility tests for numbers through experimentation and testing procedures to create a process (algorithm) to test any number for divisibility.

This task can be approached in two ways, depending on the student. Students may be provided with the divisibility tests up to a certain number and asked to construct a flowchart or decision tree demonstrating the process for testing the divisibility of any number. Alternatively, students may be required to investigate and find these test results before creating their divisibility testing algorithm.

**Learning objectives:** Students can test a number for divisibility using an algorithm.

**Resources:** Provide each small group (two or three students) with the following resources:

set of printed Divisibility rules. These rules can be found online at [Maths is fun](https://www.mathsisfun.com/divisibility-rules.html) or see [Appendix 4](#Appendix4)

butchers’ paper/A3 paper to brainstorm and present flowcharts

pens, coloured markers, pencils

calculator or computer for checking calculations.

Divide and conquer

In small groups of two or three, students explore the divisibility tests for numbers through experimentation and through testing procedures to create a process (algorithm) to test any number for divisibility.

Provide each group with the materials listed in the [Overview of the resource](#OverviewOfResource).

Activity, Part 1

Define it

Encourage students to define ‘divisibility’ within small groups or pairs. What do we mean byasking, for example,‘Is a number divisible by 4?’. What are we looking for? You may want to revise the following terms and concepts as a class.

Composite number

A non-zero natural number that has a factor other than 1 and itself is a composite number. For example, 18 is a composite number because it has the factors 1, 2, 3, 6, 9 and 18 (more than 1 and itself).

Factor

A natural number that divides exactly into another given natural number. For example, 2 is a factor of 12, since 2 × 6 = 12. 6 is also a factor of 12.

Prime number

A natural number greater than 1 that has exactly two distinct factors, 1 and itself.

The number 1 is *not* a prime number (it has only one distinct factor), nor is the number 8 as it has four distinct factors {1, 2, 4, 8}. The number 2 is the only even prime number.

Remainder

The amount left over when one number *a* is divided by another *b*. If *a* is divisible by *b* then the remainder is 0. For example, 66 *is* divisible by 11 because the remainder is 0 (since 66 = 6 × 11 + 0). However, when 68 is divided by 11, the remainder is 2, because 68 can be expressed as 68 = 6 × 11 + 2. So, 68 is *not* divisible by 11.

Share

Have students share their definitions with each other and with the class to ensure students understand the purpose of the activity.

**Tip:** Teachers are encouraged to use and clearly define the term ‘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 give students the opportunity to come up with their own agreed class definition before, during or after the activity, and then compare it with the definition in the glossary.

Activity, Part 2

Divide and conquer

In small groups or pairs, students will create a decision tree (or other visual ‘map’) to classify whether a number is divisible by a selection of other numbers.

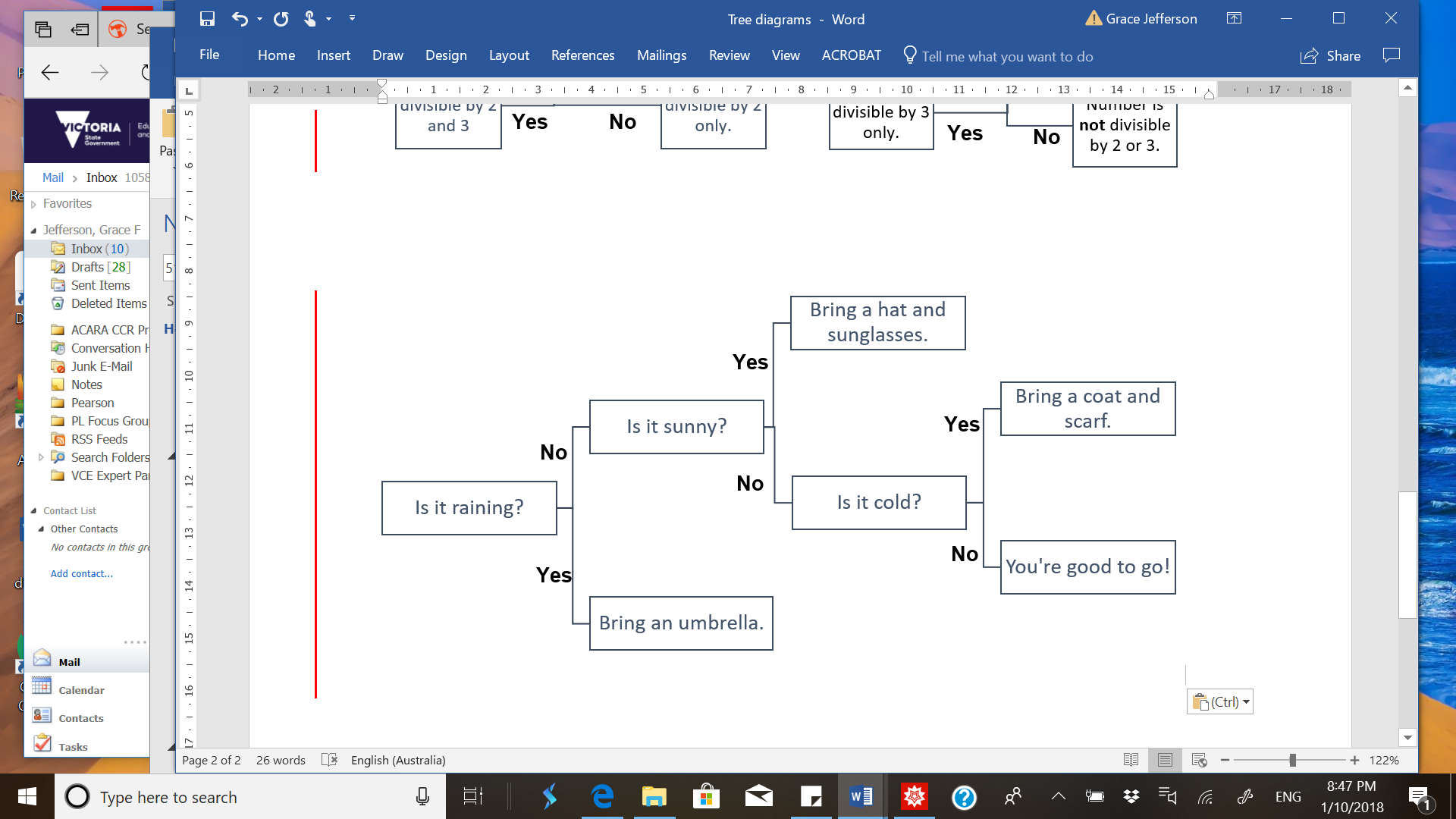
You could limit the numbers to be tested or encourage students to decide the numbers to be tested for their group or pair. For example, the number tests provided in the Divisibility test (see [Appendix 4](#Appendix4)) are for divisibility by 2, 3, 4, 5, 6, 8, 9 and 10.

**Discussion prompts**

* Do we need to check a number for divisibility by 1? Why/why not? What do we know about dividing a number by 1?
* What will be the largest number you test as a factor?
* What if a number is a prime? Does it need to be tested?

Students could investigate online to find divisibility tests themselves, or you can supply them with divisibility test rules (see [Appendix 4](#Appendix4)).

Once students have rules to use to test for divisibility, they can start to create a decision tree.

Give students an example using an everyday situation, such as what to bring if they are leaving the house in the morning, as shown in the example below.

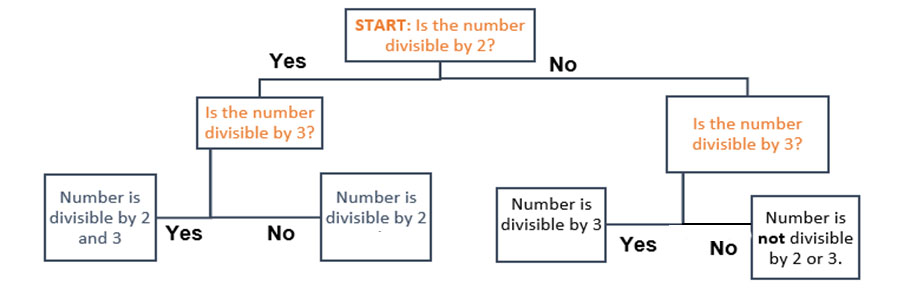
While this has been simplified (it could be rainy *and* cold so we could have joined up the rainy path to the ‘Is it cold?’ question), the decision tree uses the idea of yes/no or true/false questions that can be followed to make a decision each time this person walks out the door.

**Discussion prompt**

* Is this an algorithm?

**Answer**: Yes, because it is a process that can be carried out mechanically, using a well-defined set of instructions, to perform a task or solve a type of problem. Here, the task is getting ready to leave the house.

Have students create a tree to test for divisibility for two numbers, for example, 2 and 3. It could look like this:



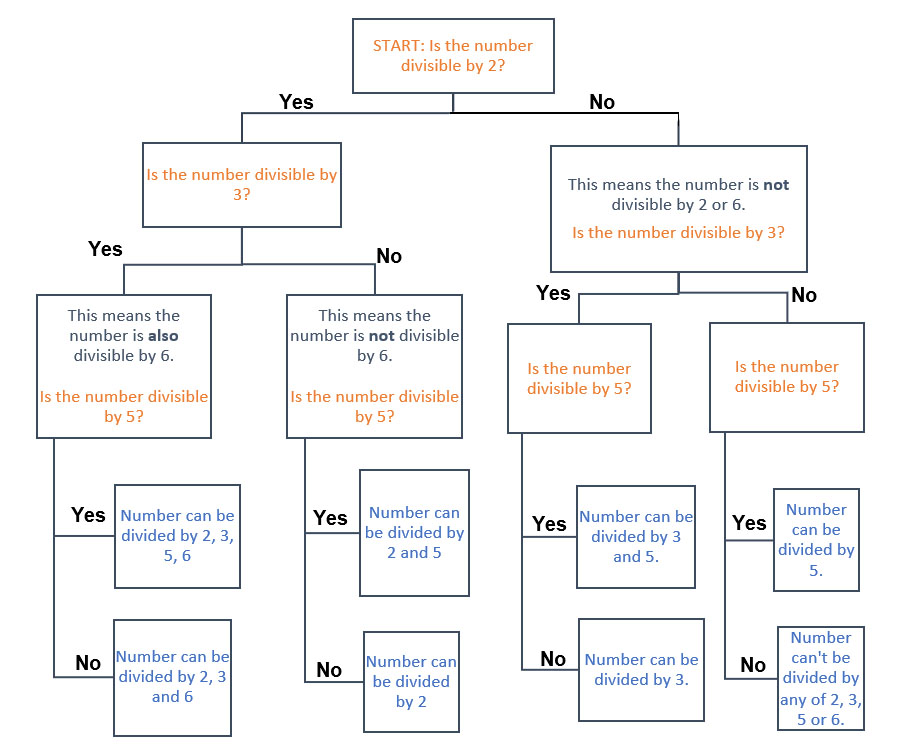
Students should test their tree. For example, in testing the tree above for the random numbers 7, 12 and 15 we would have the following sequence of answers:

7: No 🡪 No 🡪 Number is not divisible by 2 or 3

12: Yes 🡪 Yes 🡪 Number is divisible by both 2 and 3

15: No 🡪 Yes 🡪 Number is divisible by 3

Next, have students slowly increase the number of integers being tested as possible divisors from two numbers to three, then four and so on. The tree can get very complicated very quickly! Students will need a lot of space to design and create.

For example, if we were testing for divisibility by the numbers 2, 3, 5 and 6, a possible tree could look like the example below (with questions in orange and conclusions in blue).

The decision trees students create could have more than two options at each branch, or they could have different decision trees for different answers. Students could have one tree for even numbers and one for odd.

Encourage students to be creative as they put together a tree and remind them that they should test, reflect and modify to try to make their algorithm as efficient and robust as possible.

**Discussion prompts**

The following prompts may help groups streamline their trees.

* If a number is not divisible by 2, do you have to check if it is divisible by 10? Why/why not?
* If a number is divisible by 6, do you have to check if it is also divisible by 2 and 3, and vice versa? Why/why not? How could you use this knowledge to reduce the number of tests on a number?

Students should test their decision tree model with a set of random numbers. They can use their tree to decide on a number’s divisibility and either calculate by hand or use a calculator to check their answer. The [Random website](https://www.random.org/sequences/) can be used to generate a sequence of random integers (select the smallest and largest possible numbers to be generated).

Students should compare their solution processes with other groups. They could create a gallery walk for the whole class to see how different groups approached this 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.

Challenge and extend

Students could use their decision tree to see how fast they can determine whether a number is divisible by other numbers. Some students might even be interested in programming their tree using conditional programming (for example, using Scratch, Python or another general purpose programming language).

Reflection

Have students share their algorithms (decision trees) with other groups or with the whole class so that students can see there are many ways to approach a problem using an algorithm.

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

Ask students:

* How does your algorithm work in each activity?
* What is the purpose of the decision tree/s you have created?
* What does your algorithm do well? Why?
* What doesn’t it do so well? Why?
* What do you think was the purpose of this investigation?

Also explore the problem-solving aspect of this task. Ask:

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

**Tip****:** Reflection is an important aspect of any computational-thinking focused activity because it encourages students to consider the different aspects of the task, such as defining the problem and breaking the task down, selecting 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 they have used might transfer to other contexts.

Additional teaching resources

* An extension activity on divisibility testing can be found at [NRICH](https://nrich.maths.org/1308). This activity also explores modular arithmetic.
* An activity that tests proposed mathematical relationships and reasoning using algebra can be found at the online resource [reSolve](https://www.resolve.edu.au/algebra-addition-chain). In this activity, students use spreadsheets to investigate potential arithmetic relationships and then use algebra to identify and justify which relationships are generally true.
* An interactive applet for students to test theorems for congruence can be found online at [The National Council of Teachers of Mathematics](https://www.nctm.org/Classroom-Resources/Illuminations/Interactives/Congruence-Theorems/). Students should work in pairs to share their algorithmic approaches to proving (or not) the given theorems. (Note: requires Flash.)
* NRICH also has a range of activities that explore error, variable and error correction/identification. Some relevant activities include:
* [Secret transmissions](https://nrich.maths.org/7482): Identifying errors in secret messages using check digits and correcting messages. There is also a link to a follow-up activity at the end of the page.
* [Does this sound about right?](https://nrich.maths.org/7418) A range of prompts for students to discuss in small groups to explore error in estimation.

Appendix 1

Suggestions for explicitly teaching Digital Technologies (stimulus only)

**Curriculum area:** Digital Technologies

**Strand:** Creating Digital Solutions

**Band:** Levels 7 and 8

**Content description:** Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors ([VCDTCD042](https://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCDTCD042))

**Achievement standard (extract):** They design … algorithms incorporating branching and iterations …

**Suggestions that extend the activity:**

* Reviewing how algorithms may look when written as English statements
* Comparing the same algorithm presented:
* as a flowchart
* as English statements
* Creating a flowchart for a common task where decisions and repetition are made, for example searching for a word in the dictionary
* Creating an algorithm as English statements for a common task where decisions and repetition are made, for example entering in a class set of test scores
* Tracing algorithms to check accuracy, predicting output based on given input (desk-checking) and identifying any errors

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-7-8.docx) for Levels 7 and 8.

Appendix 2

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

**Curriculum area:** Critical and Creative Thinking

**Strand:** Questions and possibilities

**Band:** Levels 7 and 8

**Content description:** Consider how to approach and use questions that have different elements, including factual, temporal and conceptual elements ([VCCCTQ032](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTQ032))

**Achievement standard (extract):** Students prioritise the elements of a question and justify their selection.

**Suggestion that links to the activity:**

* Discussing the importance of defining key terms (for example, ‘divisibility’ and ‘factor’) in this investigation, identifying them and explaining why these should be defined first before working through the activity

**Curriculum area:** Critical and Creative Thinking

**Strand:** Reasoning

**Band:** Levels 7 and 8

**Content description:** Examine how to select appropriate criteria and how criteria are used in clarifying and challenging arguments and ideas ([VCCCTR039](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTR039))

**Achievement standard (extract):** They explain and apply a range of techniques to test the strength of arguments.

**Suggestion that links to the activity:**

* Examining why identifying criteria associated with the definition of an algorithm is useful in arguing whether a decision tree is an algorithm

**Curriculum area:** Critical and Creative Thinking

**Strand:** Meta-Cognition

**Band:** Levels 7 and 8

**Content description:** Consider a range of strategies to represent ideas and explain and justify thinking processes to others ([VCCCTM040](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTM040))

**Achievement standard (extract):** Students use a range of strategies to represent ideas and explain and justify thinking processes to others.

**Suggestion that links to the activity:**

* Identifying different ways to represent ideas, such as flowcharts, diagrams, decision trees and/or symbolically, and discussing their different applications and strengths or weaknesses (evaluation)

**Curriculum area:** Critical and Creative Thinking

**Strand:** Meta-Cognition

**Band:** Levels 7 and 8

**Content description:** Examine a range of learning strategies and how to select strategies that best meet the requirements of a task ([VCCCTM041](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTM041))

**Achievement standard (extract):** Students … evaluate the effectiveness of a range of learning strategies and select strategies that best meet the requirements of a task.

**Suggestion that links to the activity:**

* Following the completion of activities students reflect on how they completed the activities and the process of their own learning, including what their best strategies might be if they encountered other contexts (see the [Tip](#Tip) in the Reflection section)

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](https://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.

Appendix 4

Divisibility tests

A number is **divisible by 2** if the last digit is 0, 2, 4, 6 or 8.

For example:

154 is divisible by 2, because the last digit is 4.

A number is **divisible by 3** if the sum of the digits in the number is divisible by 3.

For example:

135 is divisible by 3 since the sum of the digits is 9 (1 + 3 + 5 = 9) and 9 is divisible   
by 3.

A number is **divisible by 4** if the number formed by the last two digits is divisible by 4.

For example:

216 is divisible by 4 because the last two digits in 216 are 16, and the number 16 can be divided evenly by 4.

A number is **divisible by 5** if the last digit is either 0 or 5.

For example:

465 and 370 are both divisible by 5 since their last digits are 5 and 0.

A number is **divisible by 6** if it is divisible by both 2 and 3.

For example:

132 is divisible by 6 since it is divisible by both 2 and 3.

We know that it is divisible by both 2 and 3 because we can test this using the divisibility tests for 2 and 3:

* last digit of 132 is 2, so it is divisible by 2
* 1 + 3 + 2 = 6, which is divisible by 3.

A number is **divisible by 7** if ...

See the following [page](#TestingDivisibilityBy7).

A number is **divisible by 8** if the number formed by the last three digits is divisible by 8.

For example:

5120 is divisible by 8 because the number formed by the last three digits is 120, which is divisible by 8 (that is, 120 ÷ 8 = 15)

A number is **divisible by 9** if the sum of all digits is divisible by 9.

For example:

495 is divisible by 9 since the sum of all the digits is 18 (4 + 9 + 5 = 18), and 18 is divisible by 9.

A number is **divisible by 10** if the last digit   
is 0.

For example:

230 is divisible by 10 because the last digit   
is 0.

Testing divisibility by 7

The process for divisibility by 7 requires a few steps. Follow the steps below to test divisibility by 7, and then work through the example provided.

1. Write down all the digits in the number except the last digit.
2. Take the last digit of the number you’re testing and double it.
3. Subtract this number from the rest of the digits in the original number that you wrote down.
4. If this new number is either 0 or a number that’s divisible by 7, then the original number is also divisible by 7.
5. If you can’t tell yet if the new number is divisible by 7, go back to the first step with this new (smaller) number and repeat.

Example (working in blue):

Let’s test the number 154 to see if it’s divisible by 7.

1. Write down all the digits in the number except the last digit.

15

1. Take the last digit of the number you are testing and double it.

4 is the last digit

4 × 2 = 8

1. Subtract this number from the rest of the digits in the original number that you wrote down.

15 – 8 = 7

1. If this new number is either 0 or a number that’s divisible by 7, then the original number is also divisible by 7.

Since 7 is divisible by 7, the original number 154 is divisible by 7!