Computational and   
algorithmic thinking   
in Mathematics

Unpacking the content descriptions

Level 3



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](#_Toc23338838)

[Overview of the resource 5](#_Toc23338839)

[Florence the Function Machine 6](#_Toc23338840)

[Additional teaching resources 11](#_Toc23338841)

[Appendix 1 12](#_Toc23338842)

[Appendix 2 13](#_Toc23338843)

[Appendix 3 15](#_Toc23338844)

[Appendix 4 17](#_Toc23338845)

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 3

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

**Content description:** Use a function machine and the inverse machine as a model to apply mathematical rules to numbers or shapes ([VCMNA139](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCMNA139))

**Achievement standard (extract):** Students… recognise the connection between addition and subtraction, and solve problems using efficient strategies for multiplication with and without the use of digital technology [and] continue number patterns involving addition or subtraction, and explore simple number sequences based on multiples.

**Title:** Florence the Function Machine

**Timing:** 2 lessons (approx. 100 minutes)

**Description:** Students explore number patterns and help Florence the Function Machine by identifying the operation being used between given input and output numbers. Students work to determine the rule (in words or numbers) needed to transform a given input number to the output.

The function machines used in these activities are one-step machines. The extension provided directs students to resources to explore two-step function machines.

Students use visual supports representing a function machine to show this process and highlight algorithmic thinking.

**Learning objectives:** Students can:

use simple function machines to represent a process

describe patterns in the number system

identify the operation used by a function machine.

**Printable materials:** These printable materials are included as [Appendix 4](#Appendix4):

Activity 1: Florence the Function Machine and the inverse machine

Activity 2: What’s missing?

Activity 3: What’s my rule again?

Activity 4: You make the rules!

Florence the Function Machine

Scaffold and support

Teachers are encouraged to use and clearly define the terms ‘function machine’, ‘operation’ and ‘algorithm’ to highlight to students that when they apply an operation like addition or subtraction of the same amount, this process is a set of instructions that a robot like Florence could carry out (an algorithm). See also 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.

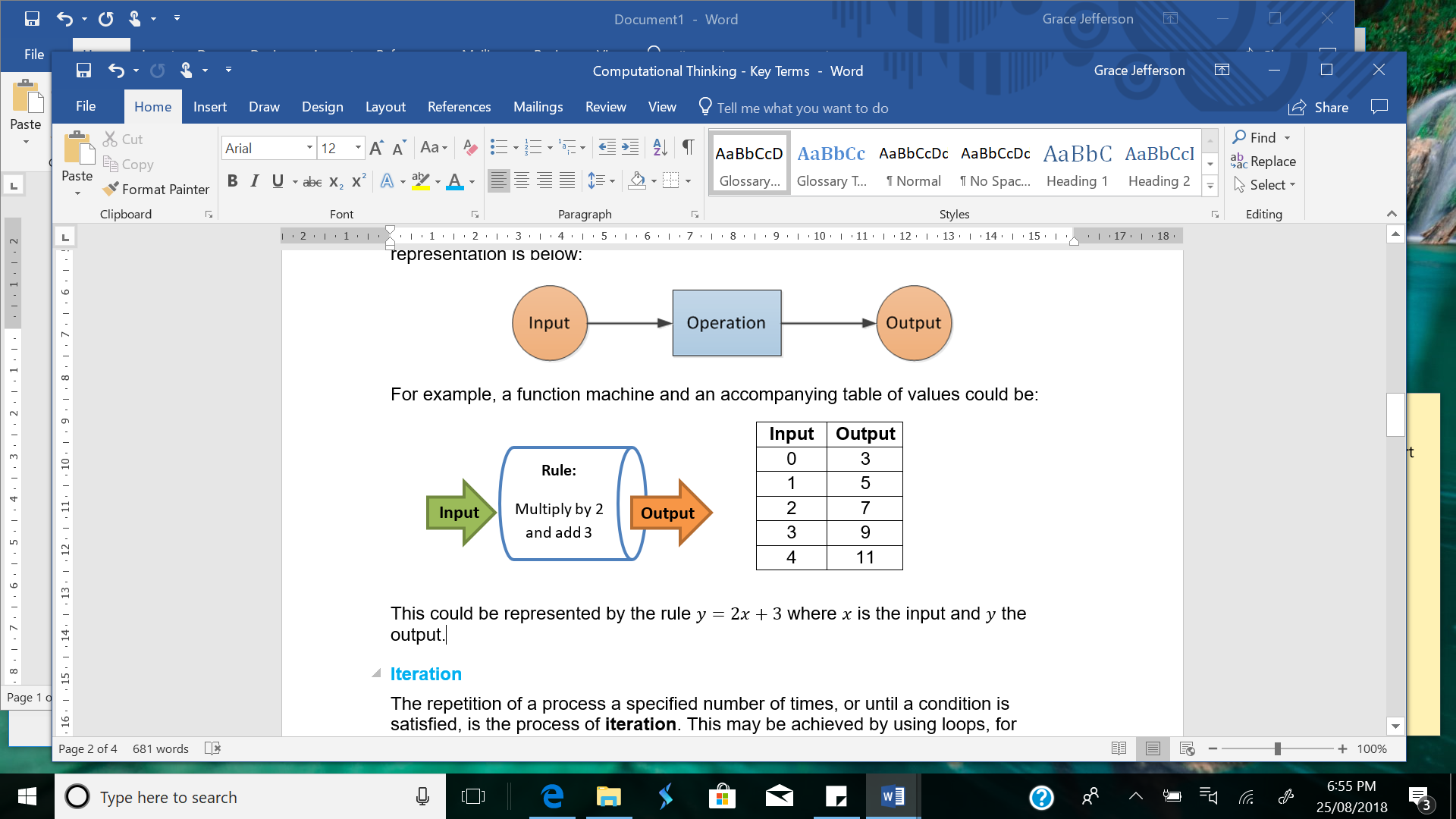
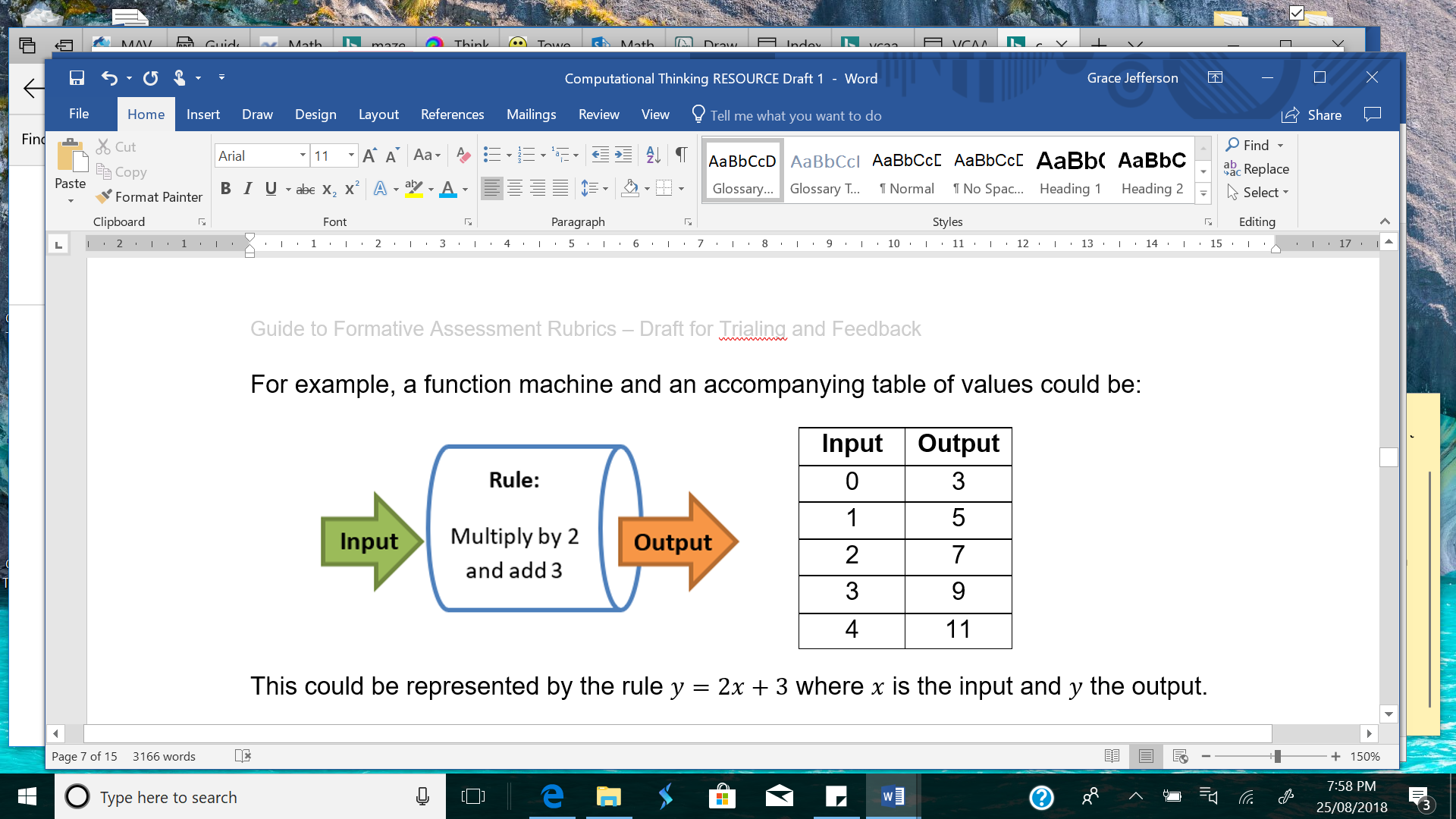
Recall that a function machine is an algorithmic process that:

* takes an input
* applies an operation (or operations)
* results in an output.

This can be displayed visually:



For example, a two-step function machine and an accompanying table of values could be:

This could be represented by the rule where is the input and is the *output.*

Teacher instructions

Students work in small groups of two or three to explore number patterns and help Florence the Function Machine by identifying the operation being used between the given input and output numbers. Students work to determine the rule (in words or numbers) needed to transform a given input number to the output.

Give each group of students a set of the printed resources (see [Appendix](#Appendix4) 4).

Activity 1 – Florence the Function Machine and the inverse machine

Ask students to work together to determine the way a rule works by taking an input (the IN number) and producing an output (the OUT number) by applying an operation to it. All input and output numbers are included, along with the rule. Students are encouraged to consider how the rule works and what the inverse function machine does compared to the original function machine. Note: These are one-step function machines.

The images of the function machine in the printable resource (see [Appendix 4, Activity 1](#Appendix4Activity1)) help students to see the process that is being used. Encourage students to use these and to refer to the idea of operations as they work.

Resources such as Unifix may also help students with this task. Encourage students to share strategies with each other.

**Tip:** Encourage students to work through the following questions or work through them together with the class.

**Discussion prompts**

* What number patterns can you see with the IN numbers for Function Machine 1?
* What number patterns can you see for the OUT numbers for Function Machine 1?
* Can you see a link between each IN number and the OUT numbers for each machine?
* What might be the rule that Florence is using to go from each IN number to each OUT number for Function Machine 1? What about Inverse Function Machine 1?
* What is different about the way Function Machine 1 and the Inverse Function Machine 1 work? What is the same?

► **Cross-curricular links**

See [Appendix 1](#Activity1) and [Appendix 2](#Activity2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Activity 2 – What’s missing?

Ask students to identify the operation used (adding or subtracting a number) in the visual representation of the action of a function machine (see [Appendix 4, Activity 2](#Appendix4Activity2)). Students use the rule provided for this function machine.

**Discussion prompts**

* What do you think the word ‘inverse’ means, after looking at how these machines work?
* What do you think an ‘inverse operation’ is?

► **Cross-curricular links**

See [Appendix 1](file:///C:/Users/anich/Dropbox/Amy/VCAA/Samples/Maths-CompThink-LevelF.docx#Appendix1) and [Appendix 2](file:///C:/Users/anich/Dropbox/Amy/VCAA/Samples/Maths-CompThink-LevelF.docx#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Activity 3 – What’s my rule again?

Provide students with an incomplete image of the function machine (see [Appendix 4, Activity 3](#Appendix4Activity3)). Ask them to fill in the missing numbers and propose a rule for each of Florence’s function machines (including the inverse function machines).

**Discussion prompts**

* What might be the rule being used in this function machine? Why do you think this?
* How do you know that your solution/rule works?
* Do you think there are other possible solutions? Why? Why not?

► **Cross-curricular links**

See [Appendix 1](file:///C:/Users/anich/Dropbox/Amy/VCAA/Samples/Maths-CompThink-LevelF.docx#Appendix1) and [Appendix 2](file:///C:/Users/anich/Dropbox/Amy/VCAA/Samples/Maths-CompThink-LevelF.docx#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Activity 4 – You make the rules!

Students create their own charts to represent the action of a function machine using the blank template (see [Appendix 3, Activity 4](#Appendix4Activity4)).

Ask students to come up with a rule for each of the function machines in the blank template and to fill in numbers that would work with this rule, or students could create a rule and fill in numbers for the IN column only and give the template to another student to complete.

**Discussion prompts**

* What strategies help you in creating a rule?
* How can you check (verify) that your rule and function machine inputs and outputs are correct?
* Is this the only possible rule for these inputs and outputs? Why/why not?

► **Cross-curricular links**

See [Appendix 1](file:///C:/Users/anich/Dropbox/Amy/VCAA/Samples/Maths-CompThink-LevelF.docx#Appendix1) and [Appendix 2](file:///C:/Users/anich/Dropbox/Amy/VCAA/Samples/Maths-CompThink-LevelF.docx#Appendix2) for ways to link this activity to the explicit teaching of Digital Technologies and/or Critical and Creative Thinking.

Challenge and extend

After creating their own function machine rules, students can move to practising their skills or even considering two-step function machines.

Students could work individually, or with others, playing with online function machines for both one- and two-step problems at [Top Marks](https://www.topmarks.co.uk/Flash.aspx?f=FunctionMachinev3).

The [FUSE website](https://fuse.education.vic.gov.au/Resource/LandingPage?ObjectId=4e986680-8c49-4c90-8034-5bca44dcac0f&SearchScope=All) also has a lesson for students to work through involving two-step function machines.

Reflection

Reinforce the idea of an ‘algorithm’ with students throughout the activity and at the end. Remember, 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.

With the class, or in small groups, ask the students to discuss the challenges they met and the different strategies they used to overcome these.

Have groups share their strategies as they work through each of the three activities.

Ask students:

* What was this activity about?
* What do the images help us do?
* Is there only one solution for each of these problems? Why? How do you know?
* What strategies did you use to come up with your own problems?
* How did you show algorithmic thinking in this task?

**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, selection of tools and processes, problem-solving, team work 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

* [Algebra: Number Sequences](https://www.resolve.edu.au/algebra-number-sequences), reSolve, is a sequence of lessons exploring additive number sequences.
* Number trains, ABC Education, is an online interactive number pattern lesson using number trains.

Appendix 1

Suggestions for explicitly teaching Digital Technologies (stimulus only)

**Curriculum area:** Digital Technologies

**Strand:** Creating Digital Solutions

**Band:** Levels 3 and 4

**Content description:** Define simple problems, and describe and follow a sequence of steps and decisions involving branching and user input (algorithms) needed to solve them ([VCDTCD023](https://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCDTCD023))

**Achievement standard (extract):** Students define simple problems … using algorithms that involve decision-making and user input.

**Suggestions that extend Activities 1–4:**

* Exploring examples of algorithms and step-by-step sequences
* Creating a sequence of steps used to solve a problem, using text, images or symbols
* Giving another student a set of instructions involving some decisions (yes or no conditions) to follow
* Defining algorithms in terms of input, processing and output
* Describing decisions involving branching in an algorithm based on a condition (user input) and explaining what happens if the condition is met or if the condition is not met
* Writing algorithms and testing them to ensure they function as anticipated

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-3-4.docx) for Levels 3 and 4.

Appendix 2

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

**Curriculum area:** Critical and Creative Thinking

**Strand:** Questions and possibilities

**Band:** Levels 3 and 4

**Content description:** Construct and use open and closed questions for different purposes ([VCCCTQ010](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTQ010))

**Achievement standard (extract):**  … students explain how to construct open and closed questions and use them for different purposes … Students select and apply a range of problem-solving strategies.

**Suggestions that link to Activities 2 and 3:**

* Elaborating on closed questions to transform them into open questions. For example, ‘What is the missing number?’ could be elaborated as the following open questions: What strategies am I using to find this missing number? What patterns can I see? How do I know this is the only solution? Why should I check this solution?

**Curriculum area:** Critical and Creative Thinking

**Strand:** Reasoning

**Band:** Levels 3 and 4

**Content description:** Identify and use ‘If, then…’ and ‘what if…’ reasoning ([VCCCTR016](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTR016))

**Achievement standard (extract):** They use and explain a range of strategies to develop their arguments.

**Suggestions that link to Activity 1:**

* Using the sentence starter ‘What if …’ to explore possible consequences, for example ‘What if I have only the rule and the OUT numbers? How can I find the IN numbers?’
* Using ‘What if …’ or ‘Imagine that …’ to find examples to explore the strength of solutions, for example ‘What if there were other numbers that worked here? How can I prove that there are/are not any other solutions?’
* Identifying how ‘if, then …’ reasoning can be used to move from separate pieces of information to a conclusion, for example ‘If I know the rule, then I can find the OUT numbers from the IN numbers and use the inverse function machine to go the other way.’

**Curriculum area:** Critical and Creative Thinking

**Strand:** Meta-Cognition

**Band:** Levels 3 and 4

**Content descriptions:**

* Investigate a range of problem-solving strategies, including brainstorming, identifying, comparing and selecting options, and developing and testing hypotheses ([VCCCTM020](http://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTM020))
* Examine an increased range of learning strategies, including visualisation, note-taking, peer instruction and incubation, and reflect on how these can be applied to different tasks to reach a goal [(VCCCTM019)](https://victoriancurriculum.vcaa.vic.edu.au/Curriculum/ContentDescription/VCCCTM019)

**Achievement standard (extract):** Students select and apply a range of problem-solving strategies … They practise and apply an increased range of learning strategies, including visualisation, note-taking, peer instruction and incubation.

**Suggestions that link to Activities 1–4:**

* Identifying and applying techniques to improve brainstorming, such as setting time limits or swapping notes or solutions part-way through to build on another’s ideas
* Undertaking trials to develop a hypothesis about the relationship between two variables (IN and OUT numbers) and reflecting on how hypotheses can be used to make predictions for the rule and the way the function machine and its inverse operate.
* Reflecting on how moving from the specific to the general helps us find the rule
* 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.

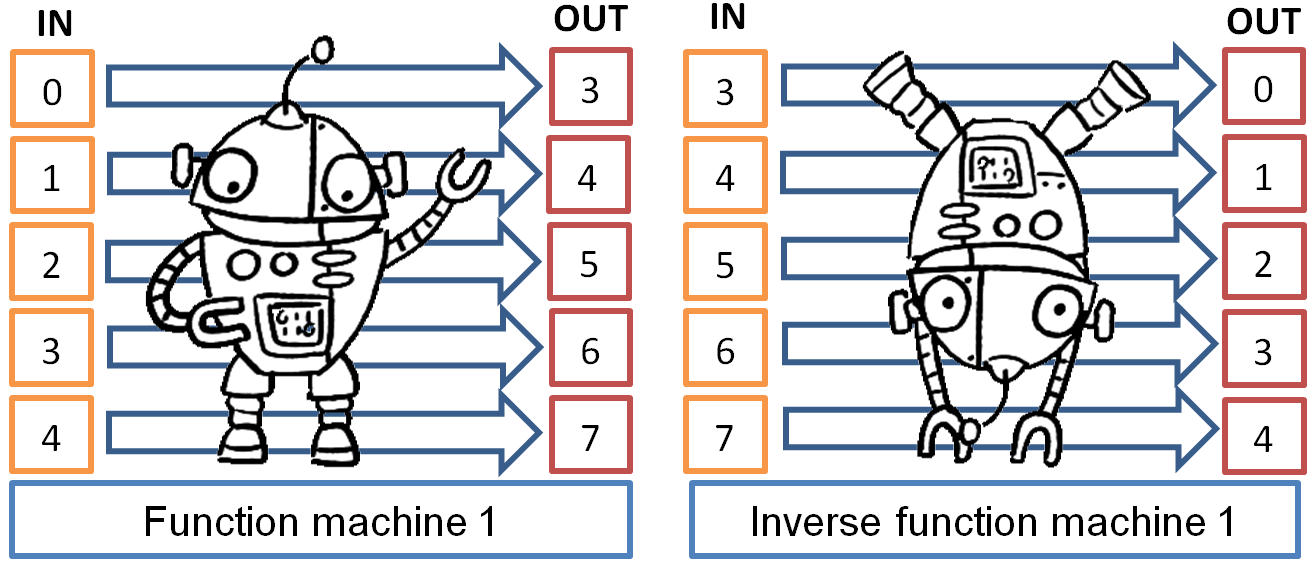
Flowcharts are often useful in visualising an algorithm.

Computational thinking

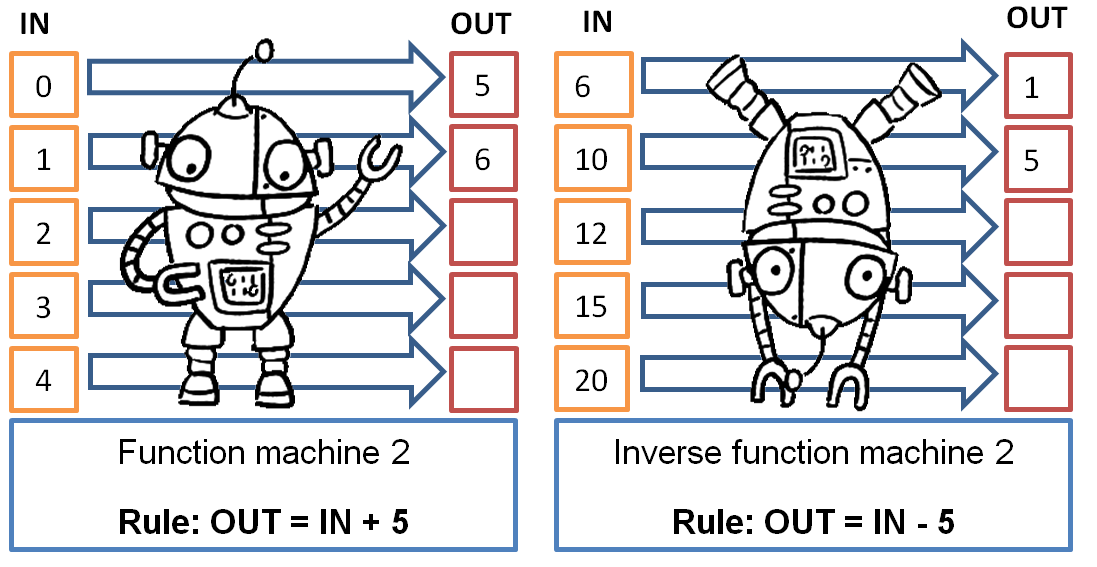
A problem-solving method that involves various techniques and strategies in order to solve problems that can be implemented by digital systems, such as organising data logically, breaking down problems into components, and the design and use of algorithms, patterns and models.

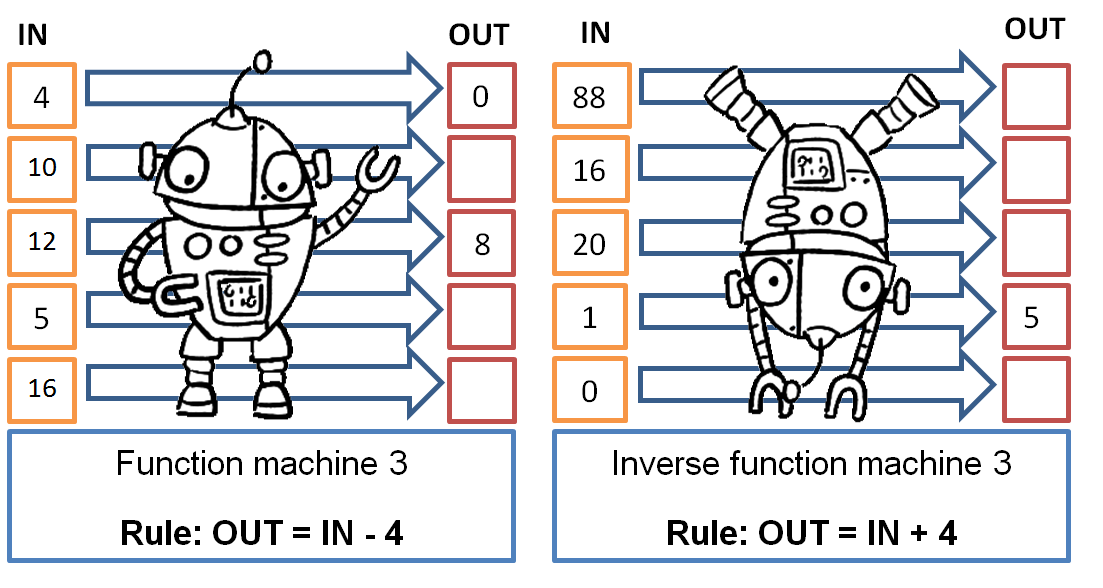
Appendix 4

Activity 1: Florence the Function Machine and the Inverse machine

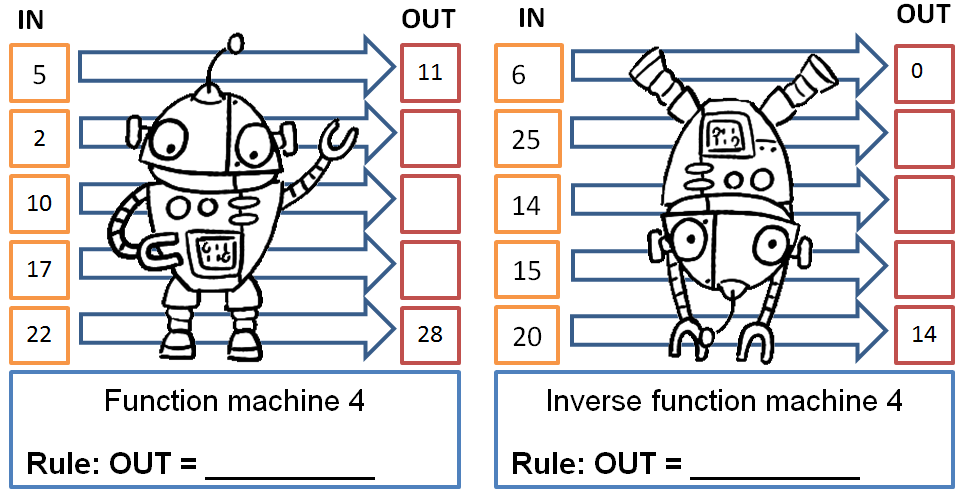


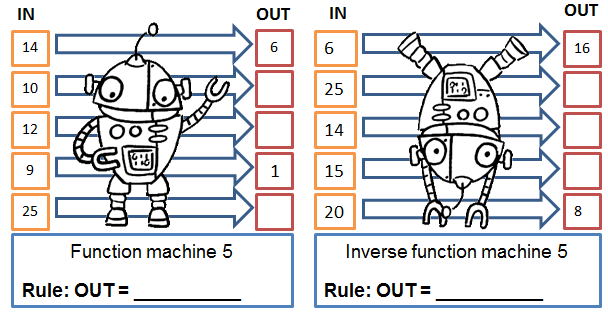
Activity 2: What’s missing?





Activity 3: What’s my rule again?





28

8

Activity 4: You make the rules!

Your turn! Come up with a rule for each function machine and fill in the numbers that would work for your rule.

Or fill in the IN numbers and give your function machines with their rules to someone else to fill in the OUT numbers.

