

Exploring Digital Technologies

A practical 30-minute session for teachers

- Using the Digital Technologies Hub effectively
 - Scope & sequence for planning
 - Adaptable learning sequences
 - Core concepts with practical examples
 - What effective DT teaching looks like
 - Flexible assessment approaches

Resources & next steps you can use right away

Walk away with practical strategies and ready-to-use resources.



Presented by Martin Richards
Curriculum and Learning Design Specialist

Acknowledgement of Country

I wish to acknowledge the Eastern Kulin Nation, Traditional Custodians of the land on which our ESA head office stands, and pay my respects to Elders past and present.

Artwork by Keisha Leon



Practical planning for DT

Wherever you're on your Digital Technologies journey, today's session should prove useful.

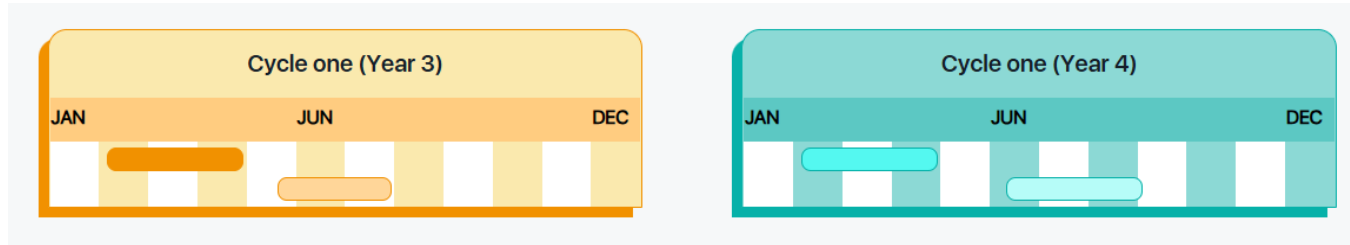
We're focusing on practical ways to plan and teach DT, with a clear focus on core concepts.

In this session, we'll focus on algorithms and do a deep dive.

I'll be using resources from the DT Hub.



Ways to timetable Digi-Tech



There are various ways to plan and implement Digi-Tech across a year.

- ▶ Short bursts within a term, using a sequence of lessons
- ▶ One dedicated term within the year
- ▶ One semester; two terms

Your chosen model will shape how you cover and revisit the core concepts.



Planning for DT Learning

Planning includes:

- ▶ Identifying the core Digital Technologies concepts
- ▶ Using practical activities that target those concepts
- ▶ Capturing evidence of learning as students engage

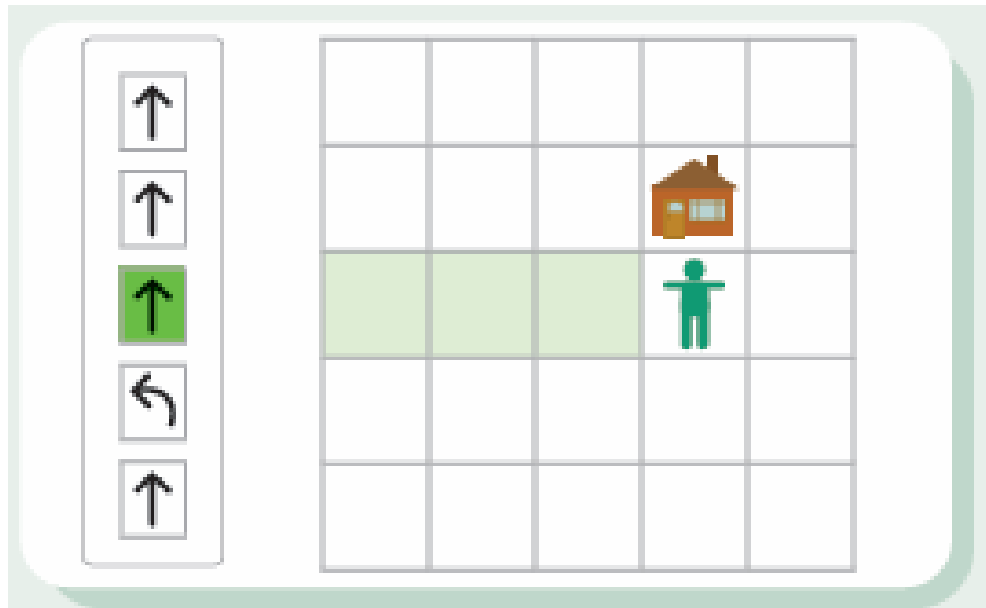
The achievement standard isn't just for reporting — it's where the core DT concepts live. Those concepts are what we plan around.



Core concept: Algorithms

Algorithms are a foundational concept across Digital Technologies:

- ▶ Work well in unplugged contexts
- ▶ Students learn to be precise
- ▶ Clear links to Mathematics, English and other learning areas



Algorithms definitions

- ▶ A clear, ordered sequence of steps and decisions needed to solve a problem.

Simplified and student friendly

- ▶ An algorithm is a set of step-by-step instructions that tell someone (or a computer) how to do something.

Why are Algorithms relevant?

Understanding algorithms helps students see how computers follow rules, make decisions, and repeat actions, forming the foundation of all programming and automation.

Core concept: Algorithms

- ▶ There are some key differences in skills required at each year band, can you spot them? And what are the common key terms?

Years 1–2

Students follow and describe basic algorithms involving a sequence of steps and branching.

Follow and describe algorithms involving a sequence of steps, branching (decisions) and iteration (repetition)

Years 3–4

Students follow and describe simple algorithms involving branching and iteration.

Follow and describe algorithms involving sequencing, comparison operators (branching) and iteration

Years 5–6

Design algorithms involving multiple alternatives (branching) and iteration

Students design algorithms involving complex branching and iteration

We're all teaching the *same three ideas* — sequencing, branching, iteration — just with increasing sophistication.

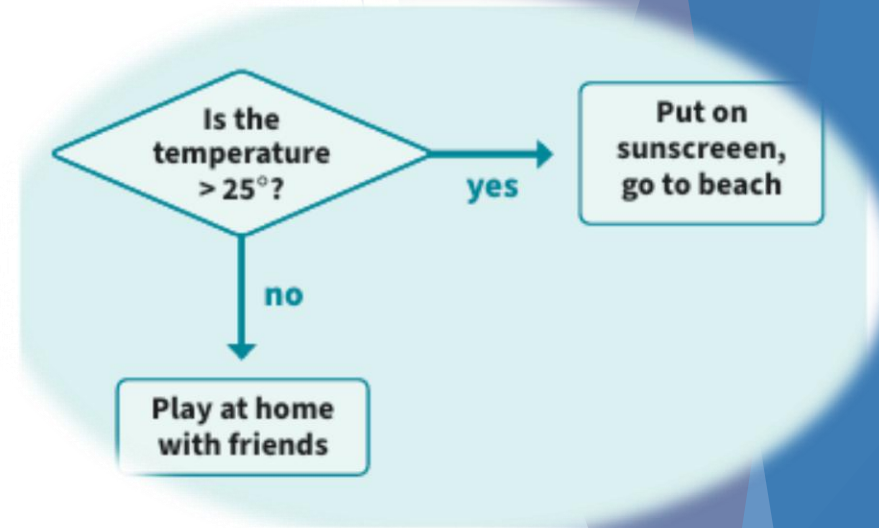
Achievement standard

Content descriptions

Using simpler terms for students

You can help students understand the ideas by using accessible language:

- ▶ Sequencing: doing things step-by-step, in order
- ▶ Branching: making a choice or deciding what happens next
- ▶ Iteration: repeating steps a number of times (a loop) until a condition is met



Algorithms Years 1-2

- ▶ Sequencing: doing things step-by-step, in order



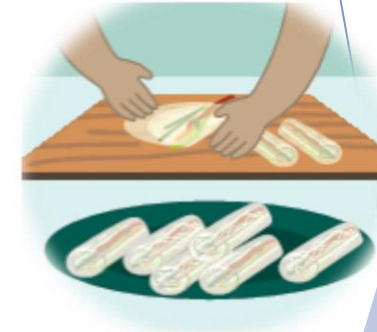
Soak the rice paper



Cut the fillings



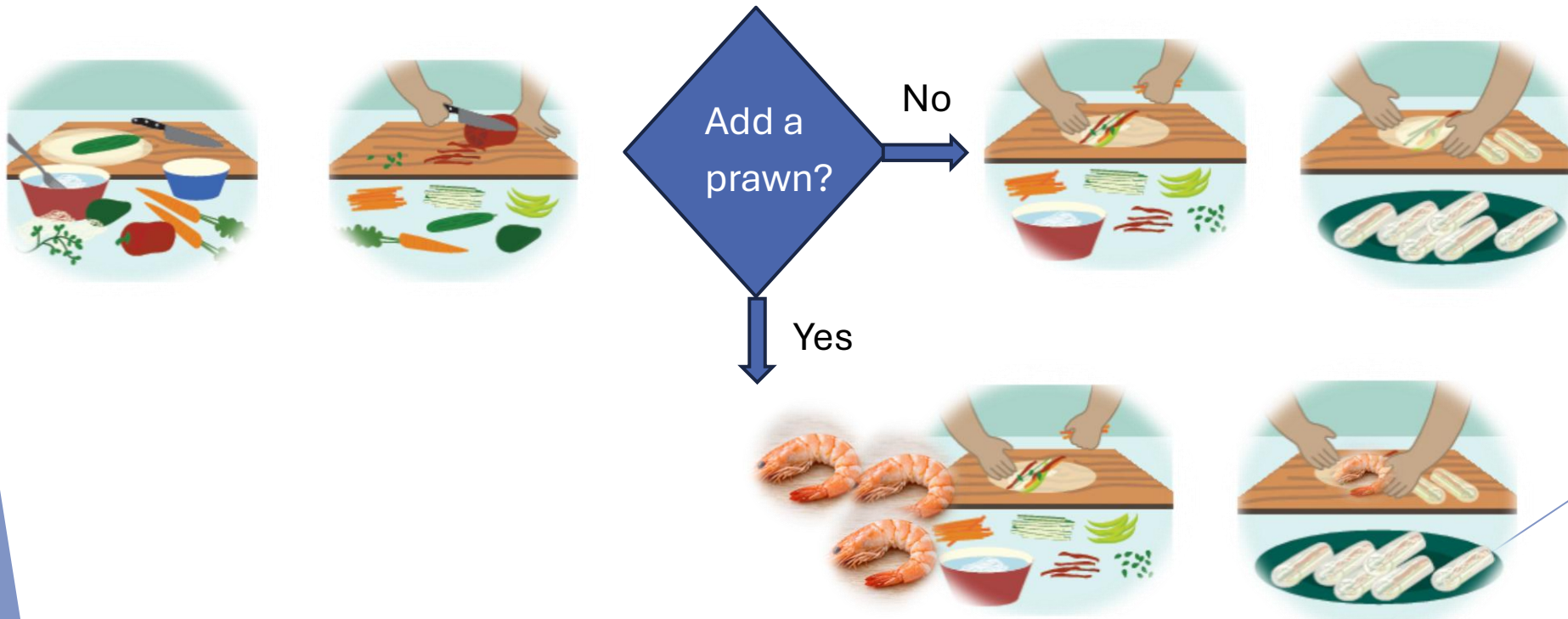
Add the fillings



Roll it up

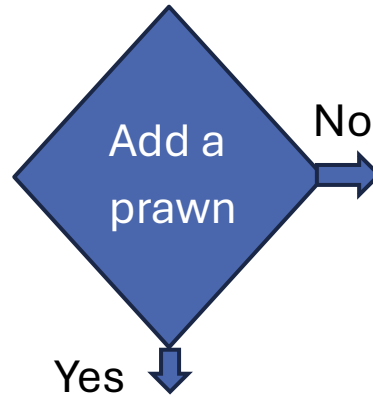
Examples using simpler terms

- ▶ Branching: making a choice or deciding what happens next



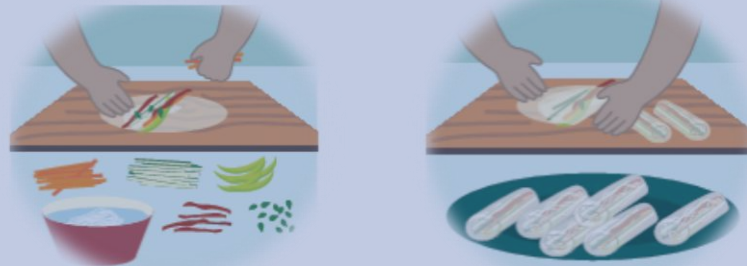
Examples using simpler terms

- ▶ Iteration: repeating steps a number of times (a loop) until a condition is met
- ▶ You need to make 10 rice paper rolls (4 vegetarian, 6 meat).

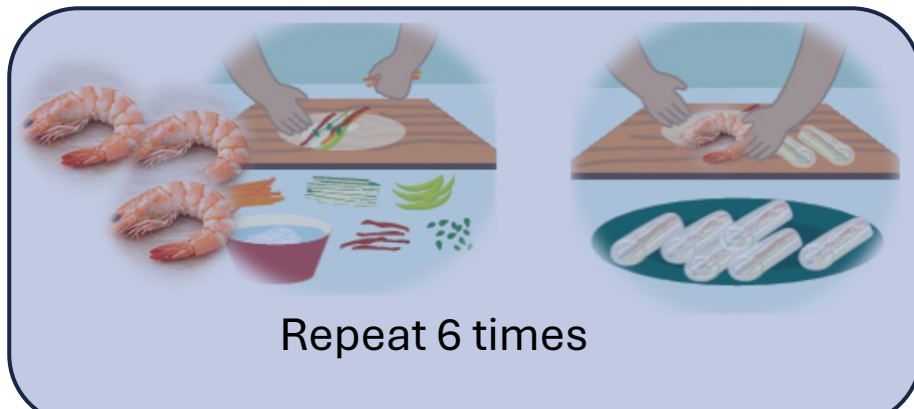


No

Repeat 4 times



Yes



Repeat 6 times

Connect the algorithm back to computers

- ▶ What we just did making rice paper rolls is exactly what computers do – they follow instructions precisely, in order.

Ask reflective questions such as:

- ▶ What would happen if we missed a step?
- ▶ What if we did the steps in the wrong order?
- ▶ We said ‘add filling’ but didn’t say *how much*. *Can we improve the instruction?*

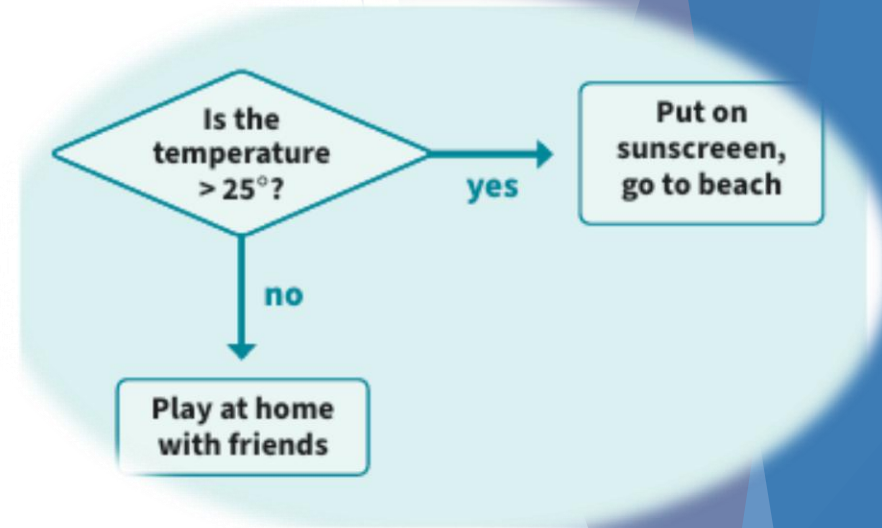
Learning focus <i>Students should be able to:</i>	Sample learning activity <i>What it might look like in the classroom</i>	Evidence of learning <i>What teachers might notice or collect</i>
<p>Follow and describe steps with choices and repetition (branching and iteration)</p>	<p>Provide simple algorithms such as picture cards or spoken instructions and order them to achieve the intended outcome. Pause to discuss where a choice is made and which steps happen again.</p>	<p>The cards are ordered correctly.</p> <p>The student responds correctly to a simple choice and describes what happens next using everyday language.</p> <p>The student repeats a set of steps and explains why they did the same steps again.</p>

Algorithms in Years 3-4

Students follow and describe the steps that include decisions based on comparing data.

Planning a day:

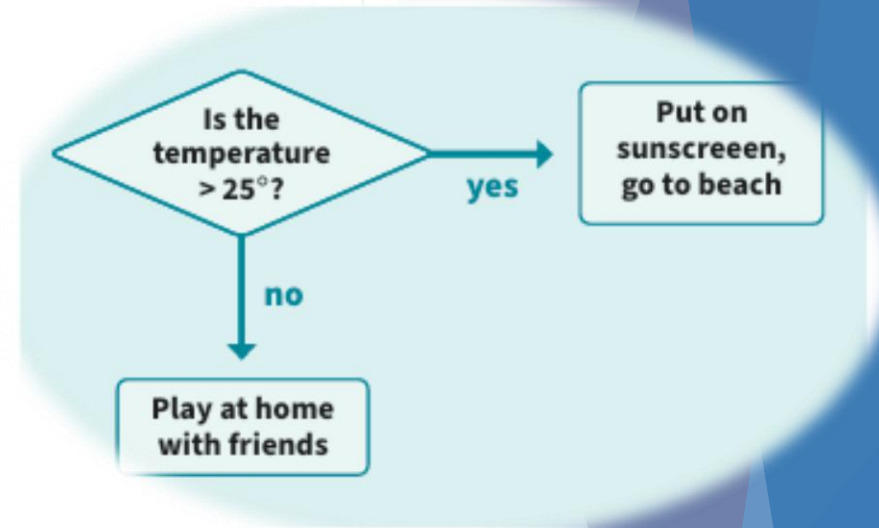
1. Wake up
2. Look outside
3. Check the temperature
4. Is it greater than 25° ?
 - Yes → Wear a hat and sunscreen, go to the beach
 - No → Stay home and play with friends
5. Return home



Algorithms in Years 3-4

Data can be anything students can check or compare – temperature, number, colour, size, time, or any information that helps them choose the next step in an algorithm on comparing data.

- ▶ Number of items: Is there more than 5 items?
- ▶ Distance: Is the object further than 2m?
- ▶ Money: Does it cost more than \$10.00
- ▶ Scores: Is the score higher than 50?
- ▶ Colour: Is the light red?
- ▶ Type/category: Does the animal have fur?

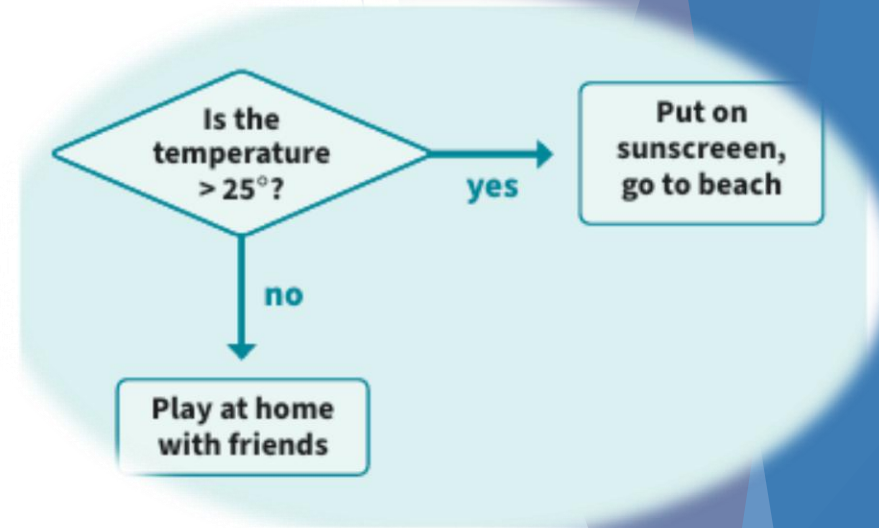


Algorithms in Years 3-4

Let's add some repeating steps (iteration)

Planning a day:

1. Wake up
2. Look outside
3. Check the temperature
4. Is it greater than 25° ?
 - ▶ Yes → Go to the beach
 - ▶ Repeat until 3.00pm
 - Swim → Get dry → Apply sunscreen → Play a game
 - ▶ No → Stay home and play with friends
5. Return home



Algorithms in Years 3-4

<p>Learning focus</p> <p><i>Students should be able to:</i></p>	<p>Sample learning activity</p> <p><i>What it might look like in the classroom</i></p>	<p>Evidence of learning</p> <p><i>What teachers might notice or collect</i></p>
<p>Follow and describe algorithms involving sequencing, comparison operators (branching) and iteration.</p>	<p>Students use number data, such as steps taken in a walking event.</p> <p>They compare these numbers using $<$, $>$, or $=$ to decide what happens next. For example, <i>if steps = 100, receive a token.</i></p> <p>They continue repeating the walking steps until they reach the target, such as 1 000 steps and earning 10 tokens.</p>	<ul style="list-style-type: none"> • The student correctly compares money amounts using $<$, $>$, or $=$. • The student follows the correct decision pathway based on the comparison made. • The student repeats the steps of the algorithm until the stopping condition is reached. • The student explains where decisions are made and why steps are repeated.

Retention, retrieval and recall

Think back to the three key ideas introduced earlier. Can you match each term with a real-world example?

For example, what do these terms mean?

- ▶ Sequencing:
- ▶ Branching:
- ▶ Iteration:



choosing a path

repeating a task

following steps in
order

Retention, retrieval and recall

- ▶ Sequencing: following steps in order
- ▶ Branching: choosing a path
- ▶ Iteration: repeating a task

Now that we've explored Years 3-4, how does this differ from what students in Years 5-6 need to understand?

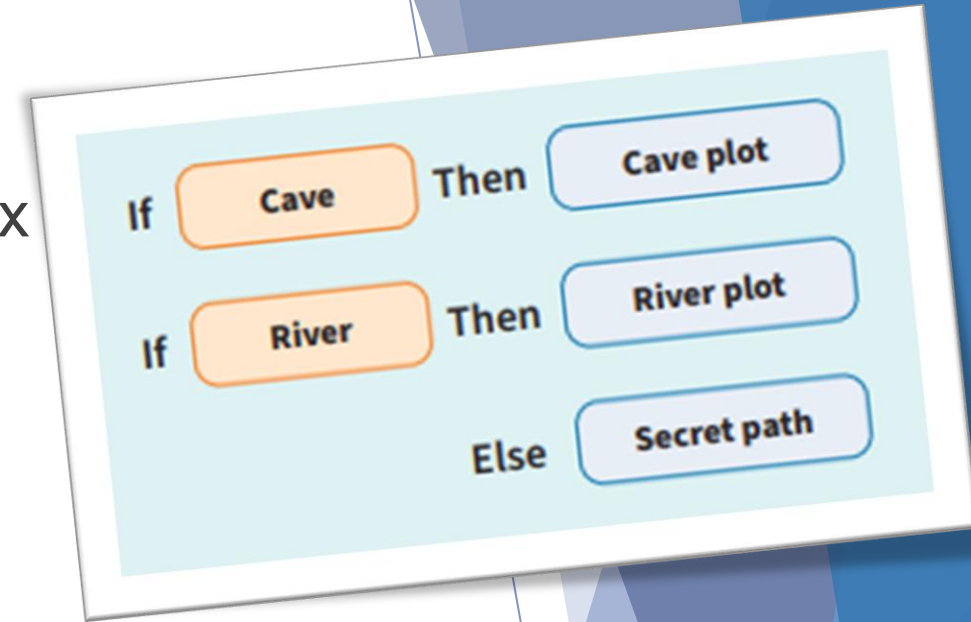


You don't need to understand every example today, just your year level.

Algorithms in Years 5-6

Students design algorithms involving complex branching and iteration.

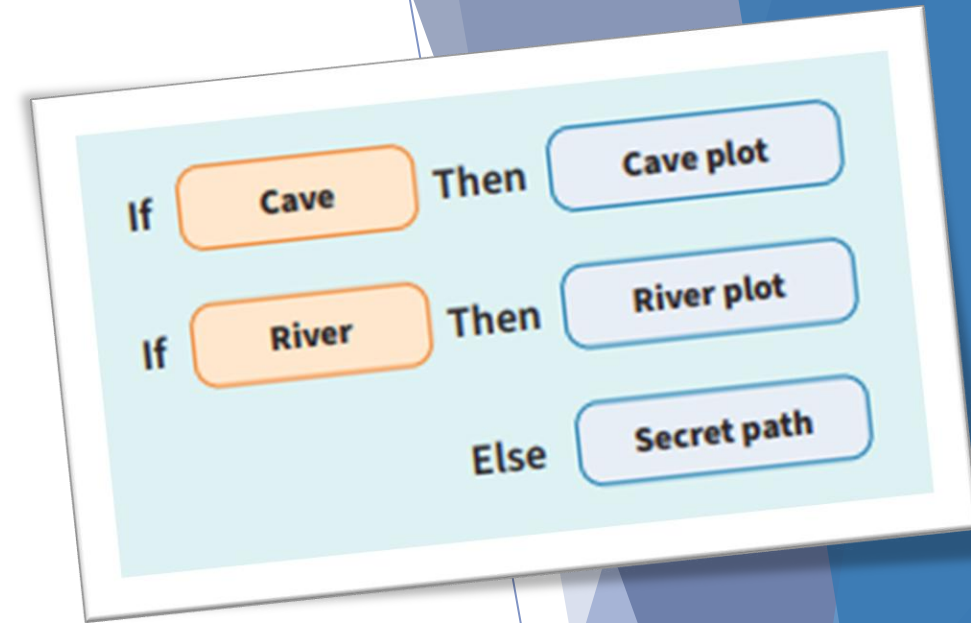
- ▶ Here's a 'choose your own adventure' scenario.
- ▶ The algorithm includes IF / THEN / ELSE statements that lead to different pathways depending on the choice made.
 - ▶ IF the choice is cave, THEN the story follows the cave plot.
 - ▶ IF the choice is river, THEN the story follows the river plot.
 - ▶ Otherwise (ELSE) the story continues to follow the secret path plot.



Algorithms in Years 5-6

Students design algorithms involving complex branching **and iteration**.

- ▶ IF the choice is cave, THEN the story follows the cave plot.
 - ▶ Enter cave
 - ▶ Repeat: Move forward → check for obstacles → Look for clues to find the treasure
 - ▶ Continue until you reach the treasure chamber and find the treasure
 - ▶ Story Ends
- ▶ IF the choice is river, THEN the story follows the river plot.
- ▶ ELSE the story continues to follow the secret path plot.
- ▶ IF the chosen pathway reaches an ending, THEN Story Complete.



Pair-test the algorithms (debugging!)

Algorithms in Years 5-6

Students turn their algorithm into a graphic novel.

Use presentation software such as PowerPoint or Keynote.

Slide 1 → Story introduction

Slide 2 → Choice buttons: “Enter Cave” or “Go to River”

Slide 3 → Cave path

Slide 4 → Cave loop (iteration)

Slide 5 → Treasure chamber ending

Slide 6 → River path

Slide 7 → Secret path (the ELSE branch)

Slide 8 → Alternate endings

Students click buttons to follow the story.

Each button takes them to a different slide – a branching algorithm.




Algorithms in Years 5-6

Students move beyond simple decisions and start working with multiple alternatives working within a forever loop. In this game example, each attempt leads to different outcomes—move back to start, lose a life, end the game.

Algorithm



To keep in the maze:

If the dragon or ghost touches the colour  , then ... move back.

If the dragon touches the treasure, then ... you win.

If the ghost catches the dragon, then ... game over.



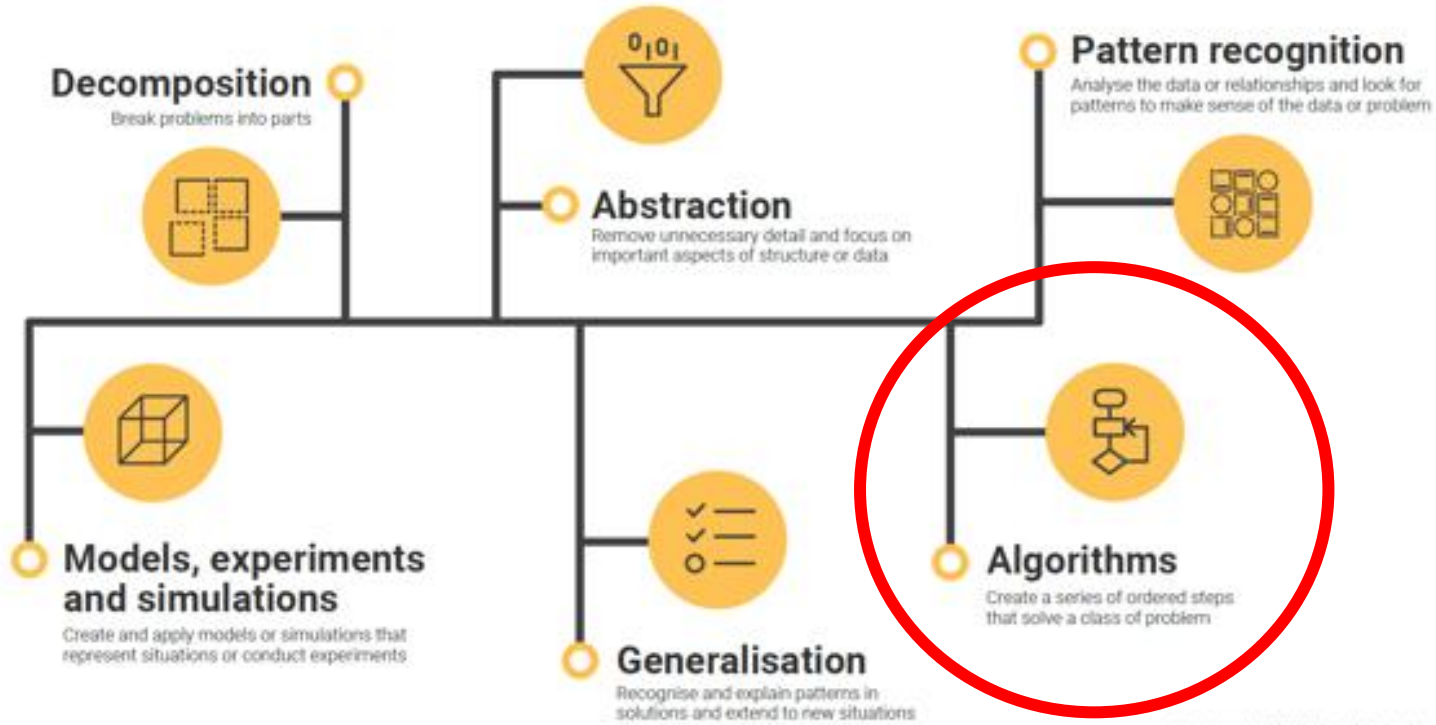
Algorithms in Years 5-6

<p>Learning focus</p> <p><i>Students should be able to:</i></p>	<p>Sample learning activity</p> <p><i>What it might look like in the classroom</i></p>	<p>Evidence of learning</p> <p><i>What teachers might notice or collect</i></p>
<p>Design algorithms involving complex branching and iteration.</p>	<p>Students design and follow an algorithm for a simple game that includes lives, decisions, and repetition.</p> <p>The game starts with 3 lives. IF an attempt is unsuccessful, THEN subtract one life. IF lives > 0, THEN the game continues. ELSE the game ends.</p> <p>Create a paper version of the game based to follow the algorithm and keep track of lives, score and how game ends.</p>	<ul style="list-style-type: none"> • The student designs an algorithm that includes clear branching decisions using if/then/else statements. • The student correctly applies iteration, repeating steps until the stopping condition is met. • The student keeps accurate count of lives and updates values correctly. • The student explains how decisions and repetition affect the length or outcome of the game. • <i>The game is programmed in Scratch</i>

Algorithms and computational thinking

Computational thinking

Computational thinking involves breaking down problems into parts; defining abstract concepts; designing and using algorithms, patterns, models, simulations and experiments; and generalising patterns, relationships and solutions.



Computational thinking is a core concept of the Australian Curriculum Technologies V9.0



Computational thinking is a mathematical process explicitly represented in the Australian Curriculum Mathematics V9.0

Developed by ACARA Digital Technologies
in focus project Australian Government
Department of Education CC BY 4.0

Core concept posters

- ▶ Algorithms are introduced early, from Year 1, and build in complexity right through to Year 10

DIGITAL TECHNOLOGIES HUB

Australian Curriculum V9.0

Algorithms

Years 1-2

The precise sequences of steps and decisions needed to solve a problem, often involving iterative (repeated) processes

ACARA, 2022

DIGITAL TECHNOLOGIES HUB

Australian Curriculum V9.0

Algorithms

Years 3-4

The precise sequences of steps and decisions needed to solve a problem, often involving iterative (repeated) processes

ACARA, 2022

Teaching about algorithms F-10

View your year band for an overview of the content related to algorithms. It also includes suggested related content so you can integrate with another learning area.

Approaches include:

- Match an algorithm to an outcome
 - Provide different algorithms and match to the task it solves, then follow.
- Arrange and follow a series of steps in correct order
 - Provide steps of an algorithm to order correctly.
- Fill in the gap of a missing step
 - Provide an algorithm with a step missing in the sequence.
- Identify elements of an algorithm
 - Which part is where we make a decision, is anything repeated?
- Compare two different algorithms for the same problem
 - Do they each work? Is one more efficient than the other?
- Fix bugs in algorithms
 - Review and modify an algorithm that includes too many assumptions, is inaccurate or is inefficient.
- Create an algorithm
 - Use computational thinking to solve a problem and create an algorithm.
- Write an algorithm in pseudocode
 - Convert algorithms from Pseudocode into code.
- Trace and validate algorithms
 - Use input data to test and check output against expected values.

Foundation

I can follow and describe ordered steps. They can include simple choices and parts that repeat.

An algorithm describes a sequence of steps and decisions. It can be spoken as instructions, written as a list, or presented as a series of images. While following an algorithm, often by physically acting out the steps, we can see if the algorithm achieves the intended outcome. For example, design and follow a series of steps and decisions to make a sandwich or rice paper roll. The order of certain steps may affect the outcome. For example, 'Slice the vegetables' is before 'Spread the ingredients onto the rice paper and roll'.

Sometimes a single step is repeated multiple times. This is called **iteration**. We can look at more efficient ways of describing repeated steps.

Branching is when the steps include a decision with a yes or no answer.

For suggested resources: <https://bit.ly/Years1and2Algorithms>

Related content

Sequence pictures to retell a story.

English: Retell and adapt familiar literary texts through play, performance, images or writing | English AC9E1E05

Related content

Follow a series of steps when designing a solution to a problem using available materials.

Design Technologies: Generate, communicate and evaluate design ideas, and use materials, equipment and steps to safely make a solution for a purpose | Design Technologies AC9TDFP01

Achievement standard

Students follow and describe basic algorithms involving a sequence of steps, branching (decision) and branching. | AC9TD1P02

Content descriptions

Follow and describe algorithms involving a sequence of steps, branching and iteration (repetition) | AC9TD1P02

Related content

Give and follow simple directions to move from one place to another using familiar reference points.

Mathematics: Give and follow directions to move people and objects to different locations in a space | Mathematics AC9M5P02

Years 3-4

I can follow and describe the steps that include decisions based on comparing data.

An algorithm can describe a sequence of steps and decisions that include comparing data. Sequencing refers to placing the steps to be performed in order, one after another. At certain steps in the sequence we may want an input to make a decision. To get a yes or no answer, branching decisions may result from a **comparison**. The **operator** may be: < (less than), > (greater than), or = (equal to).

With **iteration**, some steps can happen a set number of times.

To move along a square pathway, repeat these steps 4 times: Move forward 5 steps, then turn right 90°.

Set instrument to piano. Play these musical notes, repeat 14 times.

- E for 0.5 beat
- B for 0.5 beat
- G for 0.5 beat
- F for 0.5 beat
- C# for 0.5 beat
- G for 0.5 beat

For suggested resources: <https://bit.ly/AlgorithmsYears3and4>

Achievement standard

Students follow and describe simple algorithms involving branching and iteration. | AC9TD1P02

Content descriptions

Follow and describe algorithms involving sequencing, comparison operators (branching) and iteration | Digital Technologies AC9TD1P02

Related content

To get 64, start with 2, then repeat this step 5 times: multiply by 2.

Mathematics: Follow and create algorithms involving a sequence of steps and decisions that use addition or multiplication to generate sets of numbers; identify and describe any emerging patterns | Mathematics AC9MA09

Years 5-6

I can design algorithms with decisions that can lead to multiple outcomes.

An algorithm can describe a sequence of steps and include multiple decisions. An algorithm can be simple or complex, but it always follows a **sequence** from a starting point.

In algorithms, if/then statements allow for different paths. For example, in a 'choose your own adventure' story, changes to cave plot. IF the choice is 'cave', THEN the story changes to a river plot. OTHERWISE (ELSE), the story continues with the secret path plot.

Yes/No questions can be used as a sorting algorithm to identify one object from a group, for example, sorting a group of animals using a series of Yes/No decisions (branching).

With **iteration**, some steps may be repeated only as long as a condition holds.

- Keep heating UNTIL temperature = 22 °C.
- Keep playing a game UNTIL 3 lives are lost, keep count of lives. Subtract one for each unsuccessful attempt.

Lives = 3

For suggested resources: <https://bit.ly/AlgorithmsYears5and6>

Achievement standard

Students follow and describe simple algorithms involving branching and iteration. | AC9TD1P02

Content descriptions

Design algorithms involving multiple alternatives (branching) and iteration | Digital Technologies AC9TD1P02

Related content

Create a flow chart to determine if numbers are divisible by 2.

Mathematics: Create and use algorithms involving a sequence of steps and decisions and digital tools to experiment with factors, multiples and divisibility; identify, interpret and describe emerging patterns | Mathematics AC9MN010

Plan your teaching

- ▶ Explore sample units and lessons.
- ▶ These sample units show how Algorithms can be taught as a short sequence or extended unit at each year level.
- ▶ Years 1-2: [Solving simple problems](#)
- ▶ Years 3-4: [Introduction to programming](#)
- ▶ Years 5-6: [Programming challenges](#)

Scope and sequence (F-10)

Learning programs to support implementation

Sequenced topics that could be used in teaching the Australian Curriculum Digital Technologies curriculum to address the content descriptions of the curriculum. The Scope and sequence has been updated to support teachers to implement AC:DT V9.0.

Years F-10 topic overview

DOWNLOAD

Foundation

Years 1-2

Years 3-4

Years 5-6

Years 7-8

Years 9-10

Solving simple problems

Overview

This unit provides opportunities to explore algorithms, conceptualising algorithms as a sequence of steps or procedures for carrying out instructions to solve problems. It also incorporates representing data as part of our algorithms.

Achievement standards

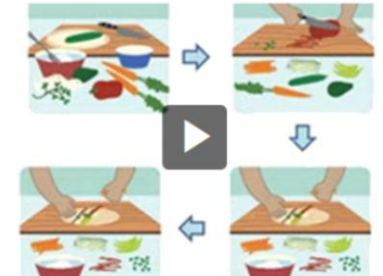
+

Australian Curriculum

+

This topic enables students to

+



Watch this video for a quick overview of the unit and how to use its resources with your students.

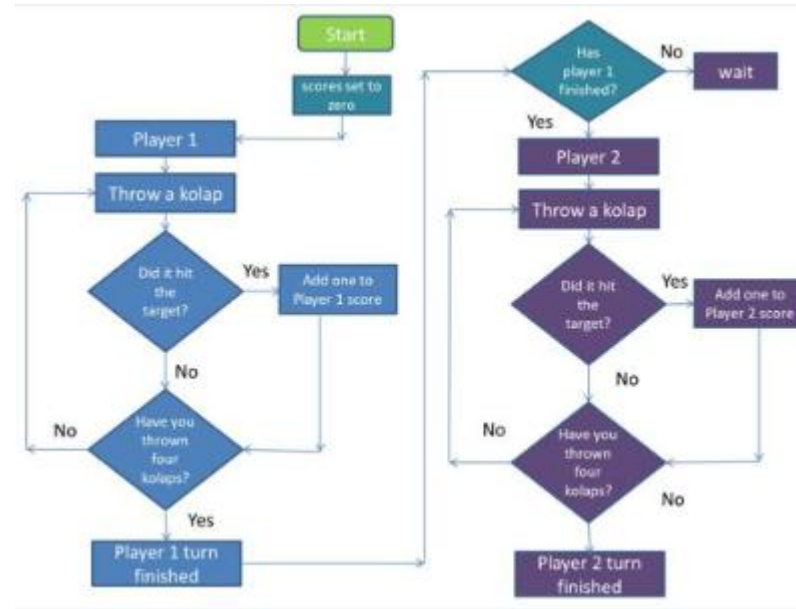
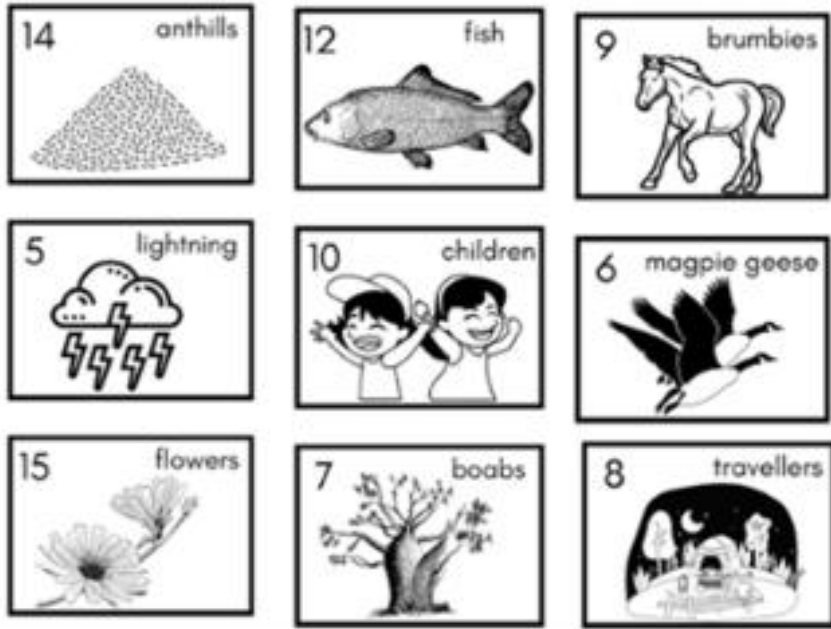
Check understanding

- ▶ What Year band does this relate to?
- ▶ Look for clues to help you.

Rubric: Algorithms

	1 (limited)	2 (basic)	3 (proficient)	4 (advanced)
Following algorithms (following verbal, written or drawn instructions)	with guidance can follow a simple sequence of steps	can follow a sequence of steps	can follow a sequence of steps with decisions (branching) and repeated steps (iteration)	can follow a sequence of steps with decisions (branching) and repeated steps (iteration), explaining the purpose of each step in achieving the overall outcome
Describing algorithms (describing an algorithm represented verbally, or as written or drawn instructions)	with guidance can describe a simple sequence of steps	can describe a sequence of steps	can describe a sequence of steps with decisions (branching) and repeated steps (iteration)	can describe a sequence of steps with decisions (branching) and repeated steps (iteration), explaining the purpose of each step in achieving the overall outcome. Can explain reasons for their choice of representation

First Nations Australians focused resources



Read a story by an Aboriginal or Torres Strait Islander author that has several clear steps in the story sequence and a connection to First Nations Australians.

Use Kolap, the instructive game that has a long history of being played on Mer Island in the Torres Strait region, to discuss how a game can be represented as an algorithm.

Learning sequence

- » Learning map and outcomes
- » Learning hook
- » Learning construction
- » Learning demo
- » Learning reflection
- » Resources

[DOWNLOAD LESSON](#)

Algorithms in Mathematics

- ▶ Algorithms feature in Number in Mathematics from Year 3
- ▶ Look for Follow and create algorithms
- ▶ Refer to Maths Hub for related resources (Year 5 example)

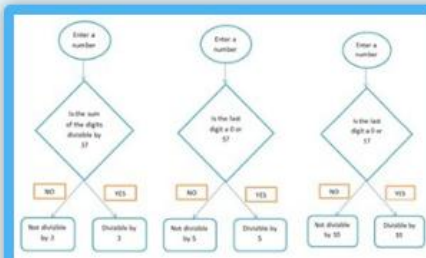
Teaching resources

	Units	Ingredient quantities	Amounts for double the ingredients
Brown sugar	grams	250	500
Self-raising flour	grams	250	500
Butter	grams	125	250
Apples		2	4
Egg		1	2
Cinnamon	teaspoons	1/2	1

Algorithms: Multiplying by a value

This lesson provides ideas for exploring contexts that multiply cells in a spreadsheet by a value.

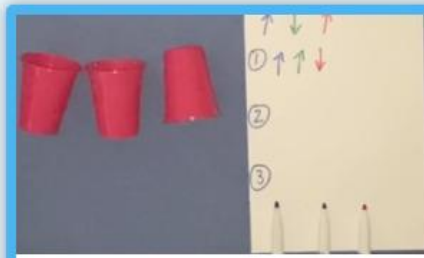
[GO TO RESOURCE](#)



Fix these divisibility rules

This lesson provides the divisibility rules as flowcharts. Students find which of the three flowcharts has an error.

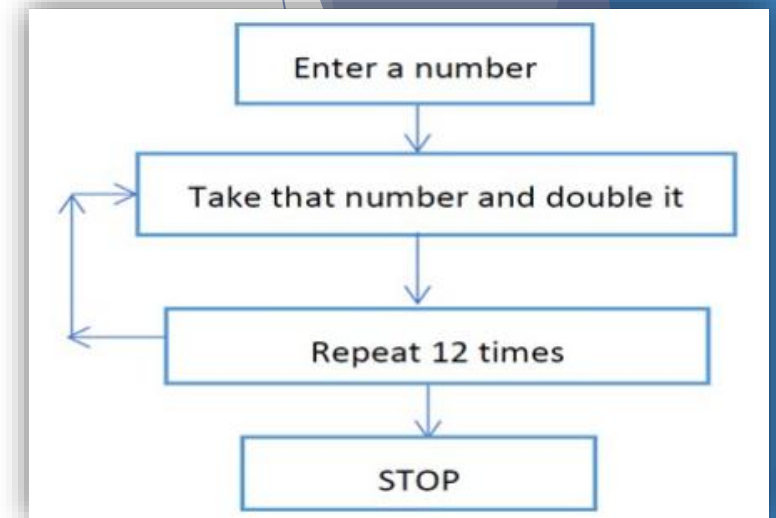
[GO TO RESOURCE](#)



Magic cups

Students apply computational thinking and create an algorithm to flip pairs of cups, in exactly three moves, to have them all facing down.

[GO TO RESOURCE](#)



Digital Technologies Hub

Find resources to help teachers, students and families learn about Digital Technologies.



Understanding DT

Engage with the Digital Technologies curriculum to plan, implement and assess effective lessons.

- [Australian Curriculum: Digital Technologies](#)
- [Core concepts](#)
- [Webinars](#)

LEARN MORE



Plan and prepare

Guidance and support to develop and implement a whole-school plan.

- [Scope and sequence \(F-10\)](#)
- [School Implementation](#)
- [Programmable robots and drones](#)

LEARN MORE



Teach and assess

Students can learn to code and create using digital technologies, and find out about relevant careers.

- [Lesson ideas](#)
- [Planning for assessment](#)
- [Guides and resources](#)

LEARN MORE



For families

Find out about Digital Technologies to support child to learn skills, digital solutions and related career.

- [Self-guides pro activities](#)
- [Learn together](#)
- [Cybersafety at](#)

LEARN MORE

Digital Technologies – Unit Planning Template (6–8 weeks)

This template is provided as a planning aid. Teachers can adapt it to suit their school context and existing planning requirements.

The intention is to use it as you explore a [scope and sequence](#) topic provided on the DT Hub.

This sample is for a Years 1-2 class focusing on algorithms. A clean template is provided after the sample.

Sample unit planning template

Year level: Years 1-2

Topic: Algorithms

Duration: 6 weeks

1. Big idea for the term

Identify one key idea students should understand by the end of the unit

Students understand that algorithms are a sequence of steps that can be followed to solve a problem or complete a task. They learn to recognise, describe and follow familiar algorithms, and begin to understand that algorithms are instructions that help computers know what to do.

2. Key skills or concepts

1. Sequence steps in order to complete a familiar task.
2. Follow and describe algorithms using images, verbally and text.
3. Begin to recognise branching (decisions) in everyday routines.
4. Begin to recognise branching and iteration (repetition) in everyday routines.

3. Weekly sequence overview

Week	Skill / concept focus	Selected activity/resource	Evidence collected
1	What is an algorithm? Recognising steps in everyday tasks Represent algorithms	Scope and sequence Years 1-2 <i>What is an algorithm?</i> (intro resource) or Sequencing Coding camp Lesson 3 (video) Students draw steps to complete a familiar activity, brush teeth, make breakfast, <u>ready</u> for bed.	Teacher observations, student drawings of step-based tasks, images sequenced in correct order
2	Sequencing steps Represent algorithms	Scope and sequence Years 1-2 Order a set of cards to create a smiley face biscuit.	Ordered picture cards, oral explanations
3	Sequencing steps Represent algorithms	Scope and sequence Years 1-2 <i>Algorithms – Storytelling sequence</i> (using a story with clear steps)	Ordered picture cards, oral explanations
4	Giving verbal instructions as commands, turn left, turn right, forward and backwards.	Scope and sequence Years 1-2 Instructing a human robot/floor robot.	Teacher observations, student steps to guide a floor robot to a position on a grid
5	Branching (simple decisions) and iteration (repetition)	Scope and sequence Years 1-2 Use traditional Australian First Nations game to introduce branching (Did you hit the target? If yes score a point; if no, then no point). Introduce repetition (throw the bean bag at the target; repeat 5 times).	<u>Refer</u> to tally of each group scores, students can describe decisions and repeating parts of the algorithm to play the game.
6	Branching (simple decisions) and iteration (repetition)	Scope and sequence Years 1-2	Complete map with arrows to direct the wolf. Student