Introducing Algorithms

Learning hook

Writing a sequence of instructions (algorithms) – whole class activity

1. Ask students to consider everyday tasks with a sequence of steps (for example, preparing to play a game using PlayStation, making a sandwich, fixing a puncture on a bicycle tyre, making a cake).
2. Ask students to suggest a sequence of steps for cleaning teeth. Remind them to sequence the steps in a logical order.
3. Share the [Tellagami](https://tellagami.com/gami/5Q4X9R/) example (https://tellagami.com/gami/5Q4X9R/) with the class. Explain that this is a fast, easy and fun way to share information.
4. Rearrange the sequence of steps for cleaning teeth. Support students to understand why order is important when undertaking some tasks. Ask students if they can identify tasks where the order of instructions is very important.

Provide the class with the steps involved in making a vegemite sandwich (out of correct order).  Ask students to arrange the instructions in the correct order.

	* Spread vegemite across the surface of the bread
	* Cut the sandwich in half
	* Take two slices of bread from the packet
	* Spread butter across the surface of the bread
	* Place the two slices of bread together



Provide multiple means of representation:

You could use printed pictures, plastic toys, or real items to help scaffold this task and support students with limited vision.

For students with no vision and who can read braille, provide braille versions of each step or provide text to speech option.

1. Give the class a list of everyday tasks. Ask students to select one task and independently write the steps necessary to complete the task.

Examples include:

	* Locate and download an app
	* Create and name a new folder on the desktop
	* Make a paper plane
	* Wash the dog
	* Fix a bicycle tyre puncture
	* Make a cup of Milo



Provide multiple means of expression:

Students may draw the steps instead of writing the step by step instructions. The student verbalises their instructions.

You may choose to have student(s) on their own redo the Vegemite example (with which they are familiar) or ask them to do a slight variation like a ham sandwich.

**Extension Activity:**

Students may choose to share their sequence of steps with others using a digital (ICT) format. For example:

• Tellagami app

• short video

• short PowerPoint presentation

• concept mapping/ graphic organiser software

• infographic

• Voki.

Learning map and outcomes

Graph paper programming

Explain that an **algorithm** is a list of steps that can be followed to complete a task.

Share the learning intentions for the lesson. For example:

Today we will …

...design a sequence of steps (using words and symbols) for others to follow

…follow and describe algorithms

…evaluate the sequences of others to determine if the outcome was successful

Learning input

Dare to square game

1. Model the ‘Dare to square’ game.
2. Ask students to work in pairs. Provide students with a 4 x 4 grid with two black squares (Grid 1). Explain that the goal is to provide instructions to direct a partner to colour the specific squares within a grid.

For example:

**Example grid 1**
Suggestion:

	1. Move one square to the right.
	2. Colour the square
	3. Move one square to the right.
	4. Move one square down.
	5. Colour the square.

Or:

* 1. Move one square to the right.
	2. Colour the square.
	3. Move diagonally to the right.
	4. Colour the square.



Provide multiple means of representation:

You may wish to use tangible objects instead of a paper grid. For example:

* use ice cube trays as the grid and tokens as the coloured squares
* alternatively create a tactile version of the dare to square grid. Each shaded square could be represented by a square of felt and stuck in place. The grid can be made using pipe cleaners or similar.
1. Provide each student with Example grid 2. Ask them to create a set of instructions to colour the correct squares.

**Example grid 2**
2. After students complete the activity, invite them to share their instructions.
3. Display the instructions for all to view. For example:

	1. Move one square to the right.
	2. Colour the square.
	3. Move one square to the right.
	4. Move one square to the right.
	5. Colour the square.
	6. Move one square down.
	7. Move one square to the left.
	8. Colour the square.
	9. Move one square to the left.
	10. Move one square to the left.
	11. Colour in the square.
	12. Move one square down.
	13. Move one square to the right.
	14. Colour the square.
	15. Move one square to the right.
	16. Move one square to the right.
	17. Colour the square.
	18. Move one square down.
	19. Move one square to the left.
	20. Colour the square.
	21. Move one square to the left.
	22. Move one square to the left.
	23. Colour the square.
4. Explain that these instructions would take a long time to write and even longer for a larger grid with more coloured squares.
Ask students to consider an easier and more effective way to provide the instructions. For example, could they use symbols?
Support the students to consider the use of arrows.
5. Provide the following suggestion:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Move one square right  | Move one square left  | Move one square up  | Move one square down  | Colour the square  |
| C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\21E9A492.tmp | C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\786B8270.tmp | C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\BA72C1FE.tmp | C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\9522BEBC.tmp | C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\8DB4E02A.tmp |

\*The program should look like this:

|  |
| --- |
| C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\5C4079C8.tmp |
| C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\AE2DAB16.tmp |
| C:\Users\Kasia\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\CC8B3F94.tmp |

Ask students to imagine they are providing the instructions to a computer. Tell them their task is to use the arrow symbols to provide the instructions instead of the words used previously.

Provide time for each student to use the arrow symbols to create a program instructing the computer to colour the correct squares on the grid.


Learning demo

Think, pair share

1. Ask students to arrange themselves into pairs.
2. Provide each pair with [Worksheet 1: Dare to square.](https://digitaltechnologieshub.edu.au/docs/default-source/getting-started-years-3-4/introducing-algorithms/worksheet-1-dare-to-square.pdf?sfvrsn=0)
3. Explain that each pair will need to view the images on the worksheet. Using the arrow symbols, convert a sequence (algorithm) into a program for others to follow to ensure the correct squares on the grids are coloured.



Provide multiple means of expression:

Students could demonstrate understanding standing/moving and using the physical classroom space (instead of completing the task on paper). You could use masking tape, for example, to create grid squares on the carpet, or use waterproof chalk outside. A student with a vision impairment can pair up with a partner who can help to negotiate the grid as they discuss the instructions.

1. Invite students to swap programs with other groups.
2. Provide each pair with blank grids and instruct the students to follow the programs to colour the squares on each grid.
3. Ask students to create their own versions of the grids and invite other students to provide instructions to complete the tasks.
4. Introduce the command repeat. Have the students consider how they can reduce the number of steps by including ‘repeat’ in the algorithm. For example, ‘move one square to the right, colour the square, repeat’.

 **Extension Activity**

For students who are confident with the ‘repeat’ concept, start using the programming term ‘loop’ and encourage them to redo the activity however this time including ‘loops’ in their instructions.

Learning Consolidation

1. Play a game that involves physical activity and following instruction such as ‘Rob the nest’.

Rules:

1) Organise 4 teams, each team has an empty hoola hoop. A central hoola hoop is filled with 8-10 bean bags.

2) One player from each team is selected to take a bean bag from the central hoola hoop until it is emptied. They then take from another team’s hoola hoop and place it in their own hoop.

3) A player is only allowed to take one bean bag at a time, and must run back to their own hoola hoop and drop it in.

4) Players cannot guard or defend your own hoola hoop.

5) The team with the most balls in their hoola hoop at the end of the game is the winner.

1.
2. Working in pairs, students write an algorithm to show the steps to play the game. Use if/then statements when describing the action if the central hoop is emptied. Students may use a repeat or loop command.  Share the completed algorithms and follow selected examples to see if the steps can be followed to play the game.

 Provide multiple means of representation:

Instead of verbal instructions, students can read and follow instructions from a card, one instruction per card. The cards can be based on a class-developed successful algorithm. Once the game has been played through several times change the order of the cards. Discuss the importance of order when writing a program (set of instructions) to follow.

1. Lead students to understand the importance of making sure instructions are clear and explicit enough to be followed. This could be a timed activity with the winning pair being those students who were able to retrieve the objects from the nest first or in the least amount of time.

Learning reflection

Encourage students to reflect on the task and share thoughts.
Highlight the positive behaviours observed during the lesson and revisit learning intentions.