Monster Maker!

Lesson Plan

**Time:** 1 - 2 hours depending on grade level.

**Audience:** This activity is suitable for students in years 1-4.

**Assumed knowledge:** N/A

**Lesson Overview**

This lesson plan accompanies our Monster Maker! short course. Students write their first programs and draw fun monster characters. Along the way, they are introduced to the concept of **algorithms** as a sequence of instructions to be followed in a specific order.

**Learning Objectives**

By the end of the lesson students will be able to:
- Define the term **algorithm**;
- Describe and follow a sequence of steps (algorithm) to solve a simple problem;
- Recognise that breaking down (or **decomposing**) a problem into smaller steps helps to solve the problem;
- Recognise that the instructions in an algorithm need to be precise;
- Recognise that the order of instructions is important and may impact the outcome;
- Debug algorithms;
- Recognise that problems may have more than one solution;
- Write programs with simple algorithms using drag-and-drop blocks.

**Lesson preparation**

**Requirements**

This activity requires an internet connection and a [supported web browser](#).

**Registration (optional)**

Registration is not required for this activity. However, unregistered accounts will expire after 24 hours, after which all student progress data is lost. [Registration](#) is free for students and teachers.
Lesson plan

Introduction - what is programming?

Start by asking students if they know what it means to program a computer?

To program a computer means to give it instructions to follow. Another word for a sequence of step-by-step instructions is an algorithm.

Giving instructions to a computer can sometimes be tricky because although computers are very fast they are not actually very smart! When we program computers we need to make sure that our algorithms are very specific so that the computer actually does what we want it to do.

Activity 1 — The Exact Instructions Challenge!

A fun way to introduce algorithms is with the Exact Instructions Challenge (https://www.youtube.com/watch?v=FN2RM-CHkul).

Discussion questions

- What algorithm did the children in the video need to write?
- What problems did they run into trying to write their algorithms, and how did they solve those problems?
Let students know that they will shortly be having a go at writing their own algorithms. This means that, like the children in the video, they will need to:

- Break a whole problem down into small steps (*decomposition*);
- Make sure the steps are in the right order;
- Make sure the steps are very precise and specific.

**New Vocabulary**

**Decomposition**: Breaking down a problem into smaller steps. Imagine if you only had the final picture to look at when building some Lego. The instructions decompose the problem into smaller steps that are easier to follow.

**Activity 2 —Write your own algorithm**

As a class, come up with an algorithm for a common activity (like pouring a bowl of cereal or brushing your teeth).

**How to brush your teeth**

1. **Step 1**: Remove the lid from the tube of toothpaste
2. **Step 2**: Hold your toothbrush in one hand, and the toothpaste in your other hand
3. **Step 3**: Squeeze the tube of toothpaste gently
4. **Step 4**:

   *Modelling an algorithm for brushing your teeth*

Use this activity to model to the class how they should set out their algorithm — start with a statement defining the problem, and then write out the algorithm as a clear sequence of instructions.
Next, have students come up with their own algorithms, individually or in pairs. Here are a few suggestions:

- Set up an obstacle course and have students write instructions to get from one end to the other, then test each other’s instructions. Remind students they need to follow the instructions exactly as they are written!
- Have students draw a simple picture, such as a stick figure or house, then write out an algorithm with steps to recreate their drawing. Without revealing their original drawing, have students test out their algorithm with other students and see how closely the result matches the original.
- Have students write technical instructions for how to perform a task of their choosing using technology, such as installing an app on a mobile device.

Activity 3 — Monster Maker!


This beginner-friendly activity introduces students to the visual programming language Blockly — students use drag and drop blocks to write and debug their own programs, and draw and colour monsters!

Activity overview

- Students read the interactive notes and solve the coding challenges by writing their own programs.
- Each block is an instruction for the computer to follow. Join the blocks together to create longer algorithms.
- Click the Run button to run and check your code. Click the Mark button to check if you are correct and see instant feedback on your solution!
**An example problem from Monster Maker!**

**Reading the interactive notes:** Students can read through the notes independently, but depending on grade and experience it might be appropriate to go through the notes as a class.

Most slides have interactive examples which can be run by clicking the ▶ button in the top right hand corner of the example. Encourage students to run these examples when they see them. If you are going through the notes as a class, this is a good opportunity to have students make hypotheses about what will happen when the program is run!

**Solving the problems:** To solve the problems, students can write their programs in the editor, run them to test that they work correctly, and then submit them for auto-marking. If it’s not correct, they’ll get instant feedback with hints on how to fix the problem.

If students run into difficulties solving the problems, encourage them to try to solve them independently. As a teacher, you also have access to Teacher’s notes in the top right hand corner of the header where you will find explanations of the solutions.

**Get creative:** The final activity is a Playground where students can further experiment with the skills they have been practising, and design their very own monster!

**Differentiation and extended learning:** When does order matter? As students are designing their own monsters in the Playground get them thinking more about
sequencing. How many **different** algorithms can they come up with which all draw the **same** monster? Sometimes problems have more than one possible solution!

**Wrap up**

Conclude the lesson by asking students to define **algorithm, program, and decomposition** in their own words and share their definitions with the class. Emphasise that the order and precision of algorithms is important to make sure that we get a correct outcome, but that sometimes a problem has more than one correct solution!

**Further learning**

Are your students keen for more coding? Here are a couple of ideas for what to try next!

- For more Monster Maker coding activities, take a look at our full-length courses **Monster Maker** and **More Monster Maker**. These courses introduce younger students to more complex algorithms, including programs with decisions and user input. ([https://groklearning.com/courses/](https://groklearning.com/courses/))
- For older students, try one of our other **free short courses**. Use Blockly to create **snowflakes**, or build your own **virtual pet**! ([https://groklearning.com/hoc/](https://groklearning.com/hoc/))

**Curriculum and standards links**

**Australian Curriculum: Digital Technologies**

- Foundation to Year 2 Processes and Production Skills: **ACTDIP004**
- Years 3 & 4 Processes and Production Skills: **ACTDIP010, ACTDIP011**

**New Zealand Curriculum**

- Technological Knowledge — Technological systems: **Level 2**

**National Curriculum in England**

- Computing: **KS 1, KS 2**

**USA CSTA K — 12 Computer Science Standards**

- **K — 2 Algorithms & Programming**: 1A-AP-08, 1A-AP-10, 1A-AP-11, 1A-AP-12, 1A-AP-14, 1A-AP-15,
- **3 — 5 Algorithms & Programming**: 1B-AP-08, 1B-AP-10, 1B-AP-11, 1B-AP-15

**USA ISTE Standards for Students**

- **Computational Thinker**