Years 3-4

Programming and algorithms and KLA examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Specific DT focus** | **Subject area** | **Short focus** | **Summary** | **Title** |
| Pathways | English | Clear, precise communication | In pairs, explore giving and following a sequence of steps and decisions to build a LEGO® toy. | Take a [LEGO® building challenge](#LEGOBUILDING) |
| Decision making (branching) | Create a storyboard to plan a story where the reader is provided with a number of decisions that lead to alternative endings. | Plan a ['choose your own adventure' story](#PLANCYOA) |
| Mathematics | Decision making (branching) | Create a flowchart to represent a sequence of (branching) steps and decisions needed to solve a mathematical problem. | Have [fun with flowcharts](#FunWithFlowcharts) |
| Choices | English | Implementing a digital solution using visual programming | Based on your storyboard for a 'choose your own adventure' story, use a visual programming language to implement a digital solution. |  |
| Create a multimodal game board where the player is provided with a number of decisions. Using Scratch and Makey Makey, add multimodal elements to the story. These are activated using an Ozobot. | [Create a game board that uses an Ozobot](#Ozobot) |
| Sequencing instructions to complete a task | Plan and create a computer program to demonstrate grammar or spelling rules, eg changing nouns from singular to plural; adding ‘ing’. |  |
| HAAS: History | Implementing a digital solution, using visual programming | Design and create a simple game/quiz to demonstrate convict crimes and punishments. | Design a quiz –[Convicts: crime and punishment](#Convict) |
| HAAS: Geography | Implementing a digital solution, using visual programming | Create a computer program for learning a traditional Aboriginal or Torres Strait Islander language. | [Create a language -learning program](#languagelearning) |
| Science | Implementing a digital solution, using visual programming | Design and create a simple quiz to explore the difference between living and non-living things. |  |
| Mathematics | Implementing a digital solution, using visual programming | Modify an existing program or create a program to design a geometric shape or design using Pencil code or similar application. |  |
| Programming a robotic device to follow a path | Create a maze or route for a programmable robot to travel. Estimate and calculate angles and distances. |  |
| Working together | HPE | Collaboration | Collaborate to decide the rules for a new game. Use a flowchart to explain the consequences of unfair play. |  |

**Title**: Have fun with flowcharts

***SUB HEADING***: Decision making (branching)

**Summary Text:** Create a flowchart to represent a sequence of (branching) steps and decisions needed to solve a mathematical problem.

**Year Level**: 3-4

Suggested steps

1. Explain how flowcharts can be used to represent algorithms involving branching (decisions). Remind students that the words, '*íf*'and'*then*’ are often used in the decision-making process (eg *If it is cold, then I will need a jacket*)*.*

To provide multiple means of [representation](https://www.digitaltechnologieshub.edu.au/teachers/inclusive-education/digital-technologies-for-all/#udl-categories), define the vocabulary and symbols that students will be using today (e.g. a diamond shape for a decision/condition, etc). You can gradually define and incorporate other related terminology such as variable and input as the lesson progresses.

1. Provide students with real-world scenarios involving decisions.
2. Demonstrate how flowcharts can be used to represent the decision-making process and ask students to create their own examples where decisions need to be made. Examples: choosing clothing items according to weather; choosing pizza toppings; using an ATM; selecting from responses provided by a phone answering machine; using a lift in a building; playing a simple game.
3. Use the lift as an example and model it, using a flowchart.  
   **Note**: The lift will only stop on a floor if the button for that floor has been pressed. For example:  
   *Start at bottom floor.   
   Go to the next floor.   
   If the button for that floor has been pressed, stop at that floor. If not, go to the next floor for which the button has been pressed.   
   Repeat until the lift arrives at the top floor.*

 To provide multiple means of expression, ask students to think about what the lift needs to do to come back down. Then, have them try to experiment with alternate ways of programming a lift, such as first determining which floors have been selected. Students could visually diagram this on paper or use a visual interface such as scratch or Microsoft Word. Further, encourage students to then write out each step in pseudocode, using programming terms such as ‘If then’, ‘While’, and ‘For’.

1. Working independently, students create a sequence of instructions to describe a simple game involving decisions. Students use flowcharts to demonstrate the algorithm (eg using dice in a game).

 You may want to provide a handout ‘cheat sheet’ or show a legend/key on the board of the different symbols and of a simple example that students can build upon.

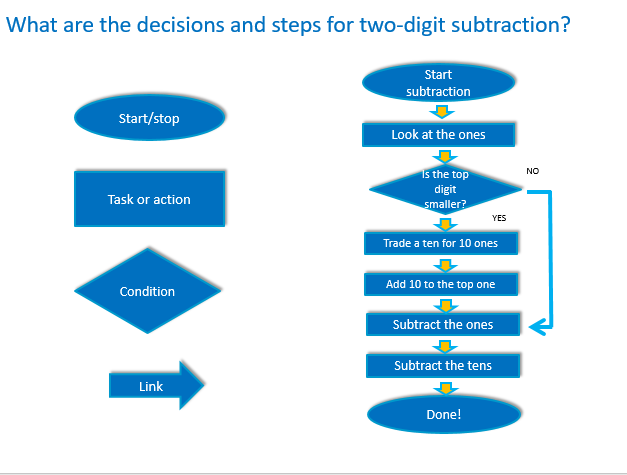


Image created by Anna Kinnane

## Challenge activity

Ask students to use a flowchart to demonstrate the steps and decisions required to:

* subtract two digits
* change a noun from singular to plural.

# Discussion

Invite students to share and compare their flowchart examples. Discuss the decisions that needed to be made in each case. For example, when changing a noun from singular to plural, what decisions need to be made? Does the word end in 'y'? If yes …

How do we know if the algorithm represented in the flowchart is correct? Explain that decision making is an important concept – choices are incorporated into a program, and computers follow these instructional pathways to deliver a desired outcome. Through discussion, ensure the students understand that computers follow instructional pathways and that these can be described using flowcharts or visual programming languages.

# Why is this relevant?

Decisions are important within computational thinking. They allow actions to be changed, based on the input of data. This input could be:

* user-input, for example selecting an onscreen value or button, typing in an answer
* sensed from the immediate environment; for example, collected via a sensor on a robotic device that senses an obstacle and is programmed to avoid it.

Algorithms are the step-by-step procedures required for solving a problem. Algorithms may be described either diagrammatically or in structured English. Flowcharts are often a good way of visualising algorithms and can be an effective way to teach the concept of ‘branching’. Branching involves making a decision between one of two or more actions, depending on sets of conditions and the data being inputted.

This activity can be used to strengthen students' understanding of computer programming as a series of instructions that can change depending on different user inputs or conditions. The focus is on how computers follow instructional pathways, and these can be described using flowcharts or visual programming languages.

# Assessment

Evaluate students understanding by assessing the completed flowcharts.

# Australian Curriculum alignment

## Technologies – Digital Technologies

Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them [(ACTDIP010)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACTDIP010)

## Mathematics

Solve simple addition and subtraction problems using a [range](http://www.australiancurriculum.edu.au/glossary/popup?a=M&t=range) of efficient mental and written strategies [(ACMNA030)](http://www.australiancurriculum.edu.au/curriculum/contentdescription/ACMNA030)

## Literacy General Capability

**Level 3**

Typically, by the end of Year 4, students:

## Grammar knowledge

**Use spelling knowledge**

* spell topic words, more complex irregular words, regular words and word families containing known letters and letter clusters, and use strategies for attempting unknown words

## Word knowledge

**Use knowledge of words and word groups**

* recognise and use nouns that represent people, places, things and ideas in the learning area and expand nouns to achieve greater precision