Digital Technologies Curriculum		Year Levels: 3-4
Descriptors and Elaborations	**Example Can Do Statements - SOLO Taxonomy** https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/3-4 https://aca.edu.au/curriculum/3-4/	**Possible Activities** (All Links work as of April 2020)
Digital Systems **#1 Identify and explore a range of digital systems with peripheral devices for different purposes, and transmit different types of data (ACTDIK007)**	Peripheral Devices (Level 3) Systems Thinking • I can identify peripherals that I use in class • I can describe what each peripheral device is used for • I can sort and classify a range of peripherals as an input/output or storage device • I can develop clues about a peripheral device, focusing on its use, and can use these clues to determine whether it is an input/output or storage device • I can complete a task that requires the use of specific peripheral devices and explain my choices • I can compare and contrast the use of peripheral devices to do a similar job such as a cabled and wireless mouse Exploring Inputs and Outputs (Level 4) Using Makey-Makey • I can identify and describe the following parts: ○ Scratch program ○ Makey Makey ○ USB cord ○ alligator clips • I can assemble the parts in the correct manner to achieve electrical flow • I can insert and change multiple sprites and sound effects which react upon receiving keystrokes • I can create a unique sound machine and evaluate its effectiveness Using Micro:bit or Codebug	• https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/3-4 • https://aca.edu.au/resources/#years-3-4 • **Create a glossary:** integrate digital technologies terms and concepts into a student created glossary. This supports students to become familiar with the language of the learning area. During lessons, make explicit use of the Digital Tech terms, reinforcing students' knowledge about what they mean and how they are used in particular contexts. Create a Word Wall where students add Digital Technology terms as they learn them. • **Put the computer together**: unplug your desktop computer and peripheral devices, including the mouse, keyboard, speakers, and printer and tell students to work as a team to connect it back together. Please note that the power should not be connected by a student. They may need to research what the different plugs are for. It may be helpful to label the different cords with their name, including VGA plug, USB plug. Remind students that they should not connect it to power without adult supervision. • **Modems Unplugged:** this activity requires students to listen to songs and find the hidden messages based on the same principle as a modem. (unplugged) • **Error Detection:** this lesson demonstrates to students how transmitting data from one computer to another can change the information. This activity shows how to detect when data has been corrupted and how to correct it (unplugged) • **How computers work:** the purpose of this activity is to give

	• I can identify and describe the following parts of a programming board: ○ LED ○ pins ○ sensors • I can describe what each component does • I can (with help), create a simple program to produce an output using some form of input • I can combine a number of components to create a desired output • I can explain how the input and output are related • I can create a program for a particular purpose and evaluate its effectiveness <u>Using LittleBits</u> • I can identify and describe the following parts of my snap-together circuit ○ LED lights ○ buzzer ○ motor (servo) ○ switch ○ power supply • I can describe what each part is used for in the circuit • I can connect some components to create light, sound or movement • I can control my device remotely • I can combine a number of components to create a desired output using different forms of input • I can create a program for a particular purpose and evaluate its effectiveness	students a sense of how computers work through role play. Students form small groups and each have a role to play as different parts of a computer. • **<u>Inside your computer</u>:** use this video to help students to understand how digital devices work around them. As an extension, ask your students to find a way to explain what they have learnt to somebody else, including making their own video, animation, writing a blog, writing a report, or creating a drawing. • **<u>Inside a Desktop Computer</u>** • **<u>Inputs and Outputs:</u>** Watch students who have made their own video to explain inputs and outputs. Get your students to creative their own video, story or animation to explain the difference between an input and output device. • **<u>Peripherals</u>** - Students will explore different types of peripherals they use every day in order to establish the data transmitted. **Equipment** • Old computers. • Beebots, Dot and Dash Robots, Edison Robots • Games consoles, new or old, which students can experiment with to explore how they work. • Makey-Makey, LittleBits, Micro:bit
Data and Information **#2 Recognise different types of data and explore how the same data can be represented in different**	<u>Secret Messages and Codes (Level 3)</u> • I can IDENTIFY a word encoded to be represented as a jumble of letters • I can IDENTIFY Morse code and braille	• **<u>Musical number patterns-the challenge</u>:** use this interactive game to support students to understand how to create patterns with a specific rule to produce the same sound pattern. Help students make connections to analysing data and understand that musical notes are data stored in a particular format.

ways (ACTDIK008)	- I can IDENTIFY a QR code Computational Thinking - I can ENCODE and DECODE a secret message using a simple way of representing the alphabet - I can SEND and RECEIVE a word using Morse code - I can WRITE and READ a word created using braille - I can WRITE A MESSAGE in Morse code and send it to a partner to be decoded - I can EXPLAIN how to send a message in Morse code - I can WRITE A SENTENCE in braille and explain how to read braille	- **Image Representation**: this activity explores how computers store data such as photos and images using numbers. - **Binary Numbers Count the dots**: this lesson plan will help students to develop an understanding of how information is stored on computers (unplugged) - **Binary Bracelets:** students create bracelets from a paper template that is a binary representation of the first letter of their name. Students learn that the same set of data can be represented in more than one way. - **Binary Baubles**: students learn about representing and storing letters in binary. They will explore how information can be stored with different combinations of just two choices of numbers 1 & 0.
Data and Information **#3 Collect, access and present different types of data using simple software to create information and solve problems (ACTDIP009)**	- I can USE AN APP to make a QR code and link this to a piece of information I created for a particular purpose. Computational and Design Thinking - I can WRITE A PROGRAM to create and send Morse code using a programming board such as BBC micro:bit Using Data to Solve Problems (Level 4) - I can IDENTIFY different classifications within a data set. - I can accurately ARRANGE data in a table with appropriate columns and rows. Computational Thinking - I can accurately ORGANISE data in a spreadsheet software and can USE that data to GENERATE a chart/graph. - I can USE software to ORDER data in a variety of different ways for different purposes. Design Thinking - I can REPRESENT the same data in a range of different ways and JUSTIFY selecting one particular way over others. - I can CREATE a presentation that makes a meaningful statement about a real-world problem, based on different	- **The Miniature Earth**: this unit plan helps to simplify world population statistics for primary aged students. It requires students to think about the data and create a display to accurately present the data they collected using a digital tool (FUSE) - **Data Representation - the NBA Store**: this unit plan supports student learning about Microsoft Excel, they produce a picture graph representing the total revenues, expenses and profit/loss margins of a sports shop (FUSE) - **We are similar but different**: this English integrated unit explores data and graphing, where students analyse different types of data and demonstrate their understanding by creating texts. - **A School Like Mine**: a sequence of lessons exploring data collection, sorting and analysis, in conjunction with the children's book 'A school like mine'. (FUSE) - **What is a database?:** use this interactive webpage to support students learning about how organisations and businesses collect and record data. As an extension activity students could investigate further an organisation / company which uses a database and report back to the class. (BBC)
Data and Information **#4 Plan, create and communicate ideas and information independently and with others, applying agreed ethical and social protocols (ACTDIP013)**		

	representations of the same set of data.	• Rubbish recording and reduction - This learning sequence is based around students surveying and collecting data concerning the rubbish that students bring in to the school each day, then using Excel to represent that data in a variety of different ways. • **Common Sense Education** ○ Digital Citizenship ○ What is your digital footprint ○ Mindful Messaging
Creating Digital Solutions **#5 Define simple problems, and describe and follow a sequence of steps and algorithms) needed to solve them (ACTDIP010)**	Introduction to Programming (Level 3) • I can define an algorithm as a series of steps • I can look at a program and identify some blocks and what they might do • I can define a problem with support • I can follow an algorithm • I can read visual programming blocks and identify some basic commands Computational Thinking • I can describe an algorithm and what each part means • I can read a program of visual blocks and describe what it might do • I can define a problem and break it into smaller parts • I can describe an algorithm for a familiar task • I can place cards of programming blocks in a sequence that may include some errors • I can explain how to create an algorithm for a simple task • I can explain what a computer program of visual blocks does • I can create an algorithm and identify where user input results in possible different actions • I can use cards of visual programming blocks to confidently create a simple program • I can follow a tutorial that uses visual programming blocks to complete a task • I can explain what the common visual programming blocks do • I can explain how to improve an algorithm	• https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/3-4 • **Hector's World**: an age appropriate animation with fun and engaging characters that explores digital safety. Children observe the characters as they explore how to stay safe online. Teachers can find full lesson plans on the website to help scaffold class discussions and follow up activities. • **Unplugged Activity: Graph Paper Programming.** • **Conditionals with Cards**: students learn about algorithms and conditional statements in this lesson. They will define circumstances when certain parts of programs should run and when they should not and determine whether a conditional is met based on criteria (Unplugged) • **'Getting up' algorithm design**: this sequence of classroom activities supports students to create a flow chart of decisions to design an algorithm. Students will use word processing or publishing software to design their algorithm of getting up in the morning and colour code it accordingly. (unplugged) • **Where is it?:** this lesson sequence has been designed to explore algorithms and programming. A student is selected to play the role of a computer and another to give explicit instructions to locate a particular number in a sequence. Students then work in pairs to practice giving and receiving explicit instructions, exploring the concepts of an effective algorithm. (FUSE)
Creating Digital Solutions **#6 Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011)**		
Creating Digital Solutions **#7 Explain how student solutions and existing information systems meet common personal, school or community needs (ACTDIP012)**		

	• I can discuss ways to improve a computer program • I can seek feedback to improve an algorithm • I can create a simple program using a visual programing language Programming Projects (Level 4) Computational Thinking • I can define an algorithm as a series of steps • I can look at a program and identify motion, control and sound blocks and describe what they might do • I can order steps in the right sequence if I'm given the steps of the task • I can identify some visual programming blocks; for example, ones for movement and making sounds • I can describe an algorithm and what each part means and indicate user input and the resulting output or action • I can read a program of visual blocks and describe what it might do • I can describe and follow a series of steps to complete a task • I can combine several blocks to create a simple program • I can explain how to create an algorithm for a task I'm going to program a solution for • I can explain what a computer program of visual blocks does and show how branching results in different actions or events • I can create an algorithm to describe a task or process • I can identify parts of the algorithm where choices are made (branching) and different events or actions result from user input or are sensed from environment • I can create a program using visual blocks and include user input and branching to allow for different options • I can explain how to improve an algorithm for example by adding branching • I can discuss ways to improve a computer program and suggest ways to debug a program if it is not working as desired • I can create an algorithm for a task and work through it and debug steps that are incorrect • I can evaluate my program, seek feedback from others and make	• **Introducing algorithms** - When teaching students basic programming skills, visual programming is the starting point. Visual programming is a way of constructing or manipulating an algorithm or program graphically rather than using text. • **How Search Works**: The life span of a Google query is less than 1/2 second, and involves quite a few steps before you see the most relevant results. Also check out the Google page on this. There are also lesson plans. • **Hour of Code** ○ Introduction ○ If/ Else Block ○ Repeat / Until Statements • **CSER Videos and Activities** **Apps and Software** • **Overview of Visual Coding Environments**: short video explaining visual programming environments and how they work by the CSER team at Adelaide University. • **Growing Up Digital Classroom Resources**: links to downloadable classroom activities, videos, interactive learning modules and advice sheets and other useful resources to use in the classroom. • **CS is Fun (free, web):** a resource bank of resources for students in age categories. Explore the different programming resources available. • **Made with Code (free, web):** this website teaches students the basics of programming while completing a project. It supports students to start thinking about digital solutions and how they could create their own creative digital solutions to solve problems. The projects use drag and drop blocks to make something work.

	changes based on feedback Communicating Ideas and Information (Level 3) • I can IDENTIFY common everyday information systems • I can IDENTIFY features and characteristics of books that allow for them to be sorted, which is part of an information system Computational Thinking • I can DESCRIBE the purpose of common information systems (eg entertainment, communication) • I can ENTER data into a spreadsheet base that allows for sorting of data by different features or elements of each item • I can ORGANISE data into a table using relevant rows and column headings that help me make sense of the data and explain how this relates to an information system Design Thinking • I can CREATE a pseudo virtual tour using a digital solution, and describe its usefulness Apply Protocols (Level 4) • I can IDENTIFY both appropriate and potentially dangerous online behaviour. Computational Thinking • I can follow a set of rules or instructions that will allow me to minimise the risk posed by potential dangerous online behaviours. • I can apply my understanding of appropriate online behavior to a variety of situations, such as fictional cyberbullying scenarios, in order to determine an appropriate course of action. • I can create a set of rules or instructions that will enable me to minimise the risk posed by potential dangerous online behaviours. Design Thinking • I can EVALUATE the appropriateness of my own behaviour and conduct. • I can work collaboratively with my peers in an appropriate way to create and complete a collaborative task/project.	• **Pencil Code (free, web):** a programming site for drawing art, playing music and creating games with block or text code. It has strong connections with maths including the areas of geometry, graphing and algorithms. Students can create using either block code to extend them they can swap to text coding using Coffeescript. Preload projects from the library or start with a blank page. • Blocky, Botlogic (for young students), Minecraft, CargoBot (iPad), Code Monkey, Gamestar Mechanic (video games), Kodable, Hopscotch, Scratch, Snap, Code Studio, Tynker, • Dot and Dash Robots, Spheros

Level 3 to Level 4 Achievement Standard

By the end of Year 4, students describe how a range of digital systems (hardware and software) and their peripheral devices can be used for different purposes. They explain how the same data sets can be represented in different ways.

Students define simple problems, design and implement digital solutions using algorithms that involve decision-making and user input. They explain how the solutions meet their purposes. They collect and manipulate different data when creating information and digital solutions. They safely use and manage information systems for identified needs using agreed protocols and describe how information systems are used.

Learning Area Achievement Standard

By the end of Year 4, students describe how social, technical and sustainability factors influence the design of solutions to meet present and future needs. They describe features of technologies that influence design decisions and how a range of digital systems can be used.

Students outline and define needs, opportunities or problems. They collect, manipulate and interpret data from a range of sources to support decisions. Students generate and record design ideas for an audience using technical terms and graphical and non-graphical representation techniques including algorithms. They plan a sequence of steps (algorithms) to create solutions, including visual programs. Students plan and safely produce designed solutions for each of the prescribed technologies contexts. They use identified criteria for success, including sustainability considerations, to judge the suitability of their ideas, solutions and processes. Students use agreed protocols when collaborating, and creating and communicating ideas, information and solutions face-to-face and online.

Years 3 and 4 Band Description

Learning in Digital Technologies focuses on further developing understanding and skills in computational thinking, such as categorising and outlining procedures; and developing an increasing awareness of how digital systems are used and could be used at home, in school and the local community.

By the end of Year 4, students will have had opportunities to create a range of digital solutions, such as interactive adventures that involve user choice, modelling simplified real world systems and simple guessing games.

In Year 3 and 4, students explore digital systems in terms of their components, and peripheral devices such as digital microscopes, cameras and interactive whiteboards. They collect, manipulate and interpret data, developing an understanding of the characteristics of data and their representation.

Using the concept of abstraction, students define simple problems using techniques such as summarising facts to deduce conclusions. They record simple solutions to problems through text and diagrams and develop their designing skills from initially following prepared algorithms to describing their own that support branching (choice of options) and user input. Their solutions are implemented using appropriate software including visual programming languages that use graphical elements rather than text instructions. They explain, in general terms, how their solutions meet specific needs and consider how society may use digital systems to meet needs in environmentally sustainable ways.

With teacher guidance, students identify and list the major steps needed to complete a task or project. When sharing ideas and communicating in online environments they develop an understanding of why it is important to consider the feelings of their audiences and apply safe practices and social protocols agreed by the class that demonstrate respectful behaviour.