Digital Technologies Curriculum Year Levels: 7-8

Content Descriptors	Example Can Do Statements - SOLO Taxonomy https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/7-8 https://aca.edu.au/curriculum/7-8/	Possible Activities (All Links work as of April 2020)
Digital Systems **#1 Investigate how data is transmitted and secured in wired, wireless and mobile networks, and how the specifications affect performance (ACTDIK023)**	Get Connected (Level 7) Systems Thinking • I can describe the types of networks. • I can describe how devices are connected in a network. • I can explain the purpose of particular devices in the network. • I can explain the advantages of having computers connected via a network. • I can explain how the internet works. • I can compare a computer network to a different type of familiar network such as a transport network. • I can describe the security measures used in our school network and their purpose. • I can describe security threats and relate the threats to certain parts of the network. • I can simulate the sending of data using connected programming boards. Networks and Performance (Level 8) • I can identify parts of my home computer network. • I can identify factors that affect network performance. Systems Thinking • I can ORDER demands on bandwidth from least to heaviest. • I can DESCRIBE speed, bandwidth, throughput and latency in relation to a computer network. • I can EXPLAIN the use of wireless and wired networks. • I can TEST the use of Bluetooth devices and gather data to	• https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/7-8 • https://aca.edu.au/resources/#years-7-8 • **Computer Chatter** - Leveraging the concept of transportation systems, students develop understandings of the properties of networked systems and the underlying techniques used to transmit and validate data. • **Encryption:** use this hands-on activity to give students a broader understanding of Internet web browser secure HTTPS communication concepts (e.g. session key's strength, problem of session key exchange). (unplugged) • **Communication over distance**: this activity is a practical session in which students work in groups to investigate the necessity of developing standards and protocols for communication using a basic electrical circuit. Lesson plans, handouts and worksheets are all included. Please note that teachers will need to create a free login to access quality teaching materials. • **How computer networks work:** Presentation and task of how computer networks work (FUSE) • **Infrastructure:** a PowerPoint presentation and activity to support students in exploring the elements of technological infrastructure. • **Signals:** this activity introduces students to communications technology giving them an understanding of vocabulary used. • **Explanation of communication systems**: students go through the different types of communication systems and learn about how they differ, with questions along the way and a student challenge at the end. (FUSE)

	answer a question about performance. • I can EXPLAIN how a smartphone is used and refer to the types of networks used for particular functions (where applicable). • I can DEMONSTRATE how to use a mobile phone as a hotspot.	• **Computer Networks explained:** this PowerPoint presentation explains different types of computer networks. (FUSE) • **How data Network works "WARRIORS OF THE NET"** - Shows basic understanding how a network data is sent across a network ,LAN, WAN and of course the World Wide Web (YouTube Video)
Data and Information **#2 Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)**	Data and Information (Level 7) Computational Thinking • I can identify parts of a problem that has been broken down into smaller components or stages • I can break down a problem into smaller parts • I can use a decision tree to break down a problem and identify requirements and constraints • I can add data to a spreadsheet or database • I can describe ways to validate data when using a spreadsheet • I can gather relevant information when using a search query in a database Design Thinking • I can collaborate with others to enter data in a spreadsheet to create a large database • I can suggest relevant column headings to organise data using a structured approach • I can validate data by creating a drop down menu in Excel • I can design a paper prototype that integrates data that is organised in a spreadsheet or database • I can evaluate the usefulness of a design focusing on the graphical user interface • I can evaluate my performance in a group project Computers and Binary (Level 8) • I can convert denary numbers to binary and vice versa. • I can IDENTIFY standard character sets.	• **Everything is Numbers** - this learning sequence looks at the nature of representation, the variety of ways data is represented in digital systems and the benefits and issues of each approach. • **See How Computers Add Numbers in One Lesson** - This video treats the adding of two numbers at the logic level, and may be helpful • **Applications of programmable systems**: students investigate one programmable system from the past which have enabled inventions of today. • **The Australian Magician's Dream:** use this lesson sequence to help students to understand computational thinking, algorithms and data. • **Computational Thinking - Searching to Speak:** use this idea to create a sequence of lessons supporting students to explore computational thinking, data, binary, and algorithms. • **Give Binary a Try:** this sequence of lessons takes students through understanding binary and how they are applied for computer engineers. • **Using binary to create on/off pictures** – students will understand how a bitmap stores images using pixels and colours them using binary numbers. • **CISCO Binary Game**: use this binary based game to help students become more proficient with calculating numbers with the binary system. • **Binary Numbers Count the dots**: this lesson plan will help students to understand how data of all kinds is stored on

	• I can IDENTIFY common audio files. Computational Thinking • I can ENCODE a grid using a red, green or blue colours that combine RGB; for example, (1,0,0) to make red or (0,0,1) to make green etc. • I can use ASCII, Unicode and Hexadecimal. • I can USE audio software to record and save an audio file. • I can EXPLAIN how RGB values for each pixel is written as 24-bit colour. • I can CODE a grid in RGB written as RGB values for each pixel written as 24-bit colour. • I can CONVERT audio files to other audio files for a particular purpose Design Thinking • I can create a website for a particular purpose and consider user needs including accessibility	computers. • **Binary Baubles:** this lesson explores the concept of binary to illustrate how a computer codes data that will be stored for later use. Students will explore computer language and how information can be stored with different combinations of • **Digital Information Unit:** what can be represented with a single bit and how do we get a single bit of information from one place to another? This unit explores the technical challenges and questions that arise from the need to represent digital information in computers and transfer it between people and computational devices. • **Text Compression:** explores with students how computers have a limited amount of space to store data and information and how they represent it. (unplugged)
Data and Information **#3 Acquire data from a range of sources and evaluate authenticity, accuracy and timeliness (ACTDIP025)**	Data and Information (Level 7) Computational Thinking • I can identify parts of a problem that has been broken down into smaller components or stages • I can break down a problem into smaller parts • I can use a decision tree to break down a problem and identify requirements and constraints • I can add data to a spreadsheet or database • I can describe ways to validate data when using a spreadsheet • I can gather relevant information when using a search query in a database Design Thinking • I can collaborate with others to enter data in a spreadsheet to create a large database	• https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/7-8 • **Computer science in a box - Unplug your curriculum:** this resource booklet was produced to be used for students ages 9-14 to teach lesson about how computers work, while addressing critical maths and science concepts such as number systems, algorithms and manipulating variables and logic. (Unplugged) • **SPARK: A better way to predict bushfires** from CSIRO. Spark will give fire-fighting agencies a more accurate view of fire behaviour, informing decisions that could minimise property damage and save lives • **Identifying High Quality Sites:** Students explore the idea that anyone can publish on the Internet, so not all sites are equally trustworthy. They need to carefully evaluate the sites they use for research, and then decide which ones they can trust. • **Datavisualization.ch** - a thoughtfully curated selection of tools
Data and Information **#4 Analyse and visualise data using a range of software to create information, and use structured data to model**		

objects or events (ACTDIP026) **Creating Digital Solutions** **#6 Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027)**	• I can suggest relevant column headings to organise data using a structured approach • I can validate data by creating a drop down menu in Excel • I can design a paper prototype that integrates data that is organised in a spreadsheet or database • I can evaluate the usefulness of a design focusing on the graphical user interface • I can evaluate my performance in a group project [Note: there is an overlap in the Can Do statements because by the SOLO taxonomy level of extended abstract (the green) the learners will have to demonstrate an abstract and deep understanding through unexpected extension]	that will make your life easier creating meaningful and beautiful data visualizations. • **CSER Videos and Activities** ○ Introduction to Data and Visualisation ○ Data Types Next Steps ○ Interview with Linda McIver : Linda McIver shares her experience implementing data and visualisation projects in the classroom using real data sources. ○ Evaluation and Assessment Part I ○ Evaluation and Assessment Part II • **General Data Visualisation Videos and Articles** ○ How to Create a Research Question ○ How to formulate a hypothesis ○ Using Critical Thinking To Find Trustworthy Websites ○ Making Charts in Google Spreadsheets ○ How To Use A Pivot Table In Google Spreadsheets ○ Poor Visualization Can Do More Harm Than Good ○ Choosing the Right Graph ○ Principles of Effective Infographic Design ○ Persuasion and the Power of Story ○ The Joy of Stats – Hans Rosling ○ How To Create Infographics
Creating Digital Solutions **#6 Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027)**	Creating an App or a Game (Level 7) Design Thinking • I can DEFINE a problem and identify functional requirements such as usability, technical or social constraints/ considerations and data requirements • I can IDENTIFY key elements by decomposing the problem. • I can interpret an algorithm presented as a flow chart and follow the steps • I can use a visual programming language or a general purpose programming language IF I copy programming	• **The Institute of Engineering and Technology:** This website provides many STEM based teaching and learning resources. For example Filer Communications and the age group that you are working with to find resources and challenges of how to teach students about networks and communication systems. • **There can be only one**- Students write a simple suite of programs that can be used to facilitate an S.R.C. election though the collection and processing of data. It assumes that students have been introduced to Python programming language.

Creating Digital Solutions **#7 Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)**	examples created by someone else Computational and Design Thinking • I can DESCRIBE two or three different design ideas and in detail discuss: ○ the logic behind transitioning between screens ○ functional requirements to judge what idea best meets these requirements. • I can use functional requirements to create an algorithm that I use to plan out a program for a digital solution. • I can create a paper-prototype of my design to show screen transitions • I can independently program a digital solution using a visual programming language • BUT I am still not confident to program using a general purpose programming language Computational Thinking • …AND I can EXPLAIN my programming choices – ○ that involve branching (where decisions by the user are enabled), ○ iteration (where loops and repeat functions have reduced the script length and detail) ○ other functions for example the use of variables. • I can independently and confidently create a digital solution using a general programming language • AND I can debug as I build. (correct my own code) Systems Thinking • AND I can EVALUATE the effectiveness of mine and other's digital solutions in meeting its functional requirements by explaining: ○ how well it meets its intended purpose ○ how the solution met one functional requirement and one constraint. • AND I can seek feedback from a small group by demonstrating my solution and then act on feedback	• Phylogenetics: Bioinformatics is a discipline that combines computer science and biology. It uses the algorithm, technology and statistics to solve problems for biology. In this activity, students participate in the process of reconstructing a phylogenetic tree and explore several core concepts making connections in biology or maths. (unplugged) • Line Drawing: computers draw lines and circles during common tasks, but how does it know which pixels to darken to create the line? This activity shows how an algorithm could be used to do it quickly. (unplugged) • Cryptographic Protocols: this activity explores Boolean logic, functions and problem solving. (unplugged) • What is a program- Understanding programming commands, Applications of programmable systems, The future of programmable systems: use this sequence of lessons to support students to build an understanding of programming and different systems available. • Computer Science in Algebra: Code.org has partnered with Bootstrap to develop a curriculum with explores algebraic concepts through programming. The 20 lessons focus on concepts like order of operations, the Cartesian plane, function composition and definition and solving problems within the context of creating a game. By the end of the course students have written programs to build a video game of their own design, which show a mastery of programming and algebra. • Introduction to Computer Modelling and Simulation: this introductory module presents basic concepts in modelling complex systems through hands-on activities and participatory simulations. A series of design and build activities guide students through developing their first computer model in StarLogo Nova. • I Love My SmartPhone mobile app development: explore the lesson plan teacher workbook and the student workbook, as well as the associated YouTube channel.
Creating Digital Solutions **#8 Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)**		
Creating Digital Solutions **#9 Implement and modify programs with user interfaces involving branching, iteration and functions in a general-purpose programming language (ACTDIP030)**		
Creating Digital Solutions **#10 Evaluate how student**		

solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)	Robotics and Embedded Systems (Level 8) Design Thinking • I can DEFINE a problem and identify functional requirements such as usability, technical or social constraints/ considerations and data requirements • I can IDENTIFY key elements by decomposing the problem. • I can IDENTIFY parts of a programming solution that uses a microcontroller • I can interpret an algorithm presented as a flow chart and follow the steps • I can use a visual programming language or a general purpose programming language IF I copy programming examples created by someone else such as a sketch of an Arduino program. Computational and Design Thinking • I can DESCRIBE two or three different design ideas and in detail discuss: ○ the 'user experience' ○ the instructions to operate the solution • I can use functional requirements to create an algorithm that I use to plan out a program for a digital solution. • I can create a paper-prototype of my design to show screen transitions • I can independently program a digital solution using a visual programming language • BUT I am still not confident to program using a general purpose programming language Computational Thinking • …AND I can EXPLAIN my programming choices – ○ that involve branching (where decisions by the user are enabled), ○ iteration (where loops and repeat functions have reduced the script length and detail) ○ other functions for example the use of variables. • I can independently and confidently create a digital solution	• **Algorithms and Programming:** students learn the fundamentals of programming constructs of JavaScript by solving problems with 'turtle graphics'. Then move on to programming simple event driven apps. • **A Beginner's Guide to Programming:** a high-level overview of some of the more commonly-used programming languages • **Survey Monkey** – create free online surveys and questionnaires, useful for collating peer and user evaluations based on criteria you establish, as well as providing options for further analysis • **Evaluation Toolbox** - though not aimed at teachers and education, you may find some useful ideas • **Introduction to Programming** by the Khan Academy • **Introduction to Genius Hour** • **CSER Videos and Activities:** Great place to learn as a teacher about all the Digital Technology aspects ○ **CSER Year 7 & 8 Project stream for Maker Space**: In this stream we will explore the use and integration of computer science and technology in creative fields, such as Art, Design, Music and Fashion! ○ **CSER Project Stream for Making Apps:** In this project stream, we will explore the design and development of a classroom project around Application development **Apps and Software (google search the software)** • **Code Combat (free, web):** the game like platform with the premise that they are stuck in a dungeon surrounded by enemies or racing across a battlefield to conquer a foe. This application puts players in control of their character through programming, using text based Java script. • **Grok Learning (free introduction to programming, paid subscriptions, web):** introductory courses using the programming language Python for people with no programming experience. Anonymous accounts disappear after 24 hours, so

	that incorporates a microcontroller and use a general programming language • AND I can debug as I build. (correct my own code) Systems Thinking • AND I can EVALUATE the effectiveness of mine and other's digital solutions in meeting its functional requirements by explaining: ○ how well it meets its intended purpose ○ how the solution met one functional requirement and one constraint. • AND I can create a 60 sec video to pitch my innovative design	students will need to sign in to save their program. • **Crunchzilla (free, web):** this online platform supports students to learn about text programming. Different options for different abilities. Code Monster is an easier version of Code Maven that offers a simple interactive tutorial and develops an understanding of text programming language. • **Python (free, computer software Windows, Mac and other):** Python is a programming language which can be used for first time or experienced programmers. There are lots of educational books to support you and your class to get started or use the helpful getting started guides on the website. For further information • **CS Field Guide (free, web):** this open sourced online textbook resource is currently mapped to the New Zealand Computer Science curriculum. There are two versions of the online textbook, the student versions and the teacher version. Teachers need to access Google groups to stay up to date with the additions and revisions and to access the most up to date link to the teacher version. Useful sections such as 'network protocols', 'data representation', and 'algorithms'. • **Bootstrap (free, web):** computing creatively and connecting mathematics. With the aim of exciting students about gaming while directly applying algebra to create something. Two main programs of learning with clear curriculum links and lesson plans for teachers. Aimed at students aged 12-16. • **Code Academy (free, web, login required):** an education platform to support students to learn programming and creating digital solutions. It is a good first step for those looking to begin more complex programming. Code Academy supports/ walks students through the steps of building a website, creating a website project or interactive or learning programming languages such as HTML and Python. • **App Inventor (free, web):** an online app developer created by MIT university. The platform is designed to support students with creating their own android apps.

		- Others ○ Code Monkey, Gamestar Mechanic (video games), Scratch, Snap, Code Studio, TinkerCAD, Hakitzu (javascript), Ruby, Looking Glass (more advanced kids – animated stories)
Creating Digital Solutions **#10 Evaluate how student solutions and existing information systems meet needs, are innovative, and take account of future risks and sustainability (ACTDIP031)** **Data and Information** **#5 Plan and manage projects that create and communicate ideas and information collaboratively online, taking safety and social contexts into account (ACTDIP032)**	Digital Citizenship (Level 7) - I can identify privacy settings on social media sites I use - I can identify rules that need to be considered and followed when connecting online - I can describe potential issues when connecting online and can describe ways to remain safe online and be responsible - I can describe how people use scams to trick others into supplying personal information - I can describe ways that copyright protects the rights of others - I can consider scenarios and explain if copyright is breached or whether there is no infringement and the use is passable Design Thinking - I can evaluate a game that presents digital dilemmas and describe its usefulness - I can create a game that presents digital dilemmas in an engaging way Connected or Distracted, Informed or Misinformed? (Level 8) Computational Thinking - I can identify types of social media - I can define social media - I can identify clickbait and fake news - I can enter data using a familiar collaboration tool with some assistance. Computational and Systems Thinking - I can describe features of social media platforms and how a user engages with these features	- https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/7-8 - **MindTools - Project Management Skills** although aimed at business, there are a range of useful tools and ideas that may provide a springboard for use with students (and not just with project management!) - **Project Management Institute Educational Foundation** – information and resources for teachers, students, parents, etc. - **eSmart Digital License:** The Digital Licence helps young people understand how to behave respectfully and appropriately online. It sets out to help young people learn about the implications of things they do online. The teacher creates an account so that the students can access it and work through the content. - **CSER Videos and Activities** ○ High Level Design – Design and Architecture ○ Design Approaches **Common Sense Education** - What is your digital footprint - Mindful Messaging

	- I can draw and label the interface of a social media platform - I can enter data into a collaborative tool and follow agreed protocols both social and technical. - I can explain some pitfalls of using social media - I can provide examples of fake news and clickbait and explain the author's intentions - I can explain how users are connected via social media - I can use collaborative tools to confidently contribute to our group's response. - I can suggest and explain the use of social and technical protocols to follow when collaborating online. Design Thinking - I can create and publish a webpage or infographic about an aspect of social media - I can evaluate the usefulness of the published information product against agreed criteria - I can help plan and manage the contribution of ideas recorded collaboratively online following agree protocols - I can provide useful feedback on other's ideas to improve the ideas presented.	

Level 7 to Level 8 Achievement Standard

By the end of Year 8, students distinguish between different types of networks and defined purposes. They explain how text, image and audio data can be represented, secured and presented in digital systems.

Students plan and manage digital projects to create interactive information. They define and decompose problems in terms of functional requirements and constraints. Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability. They analyse and evaluate data from a range of sources to model and create solutions. They use appropriate protocols when communicating and collaborating online.

Learning Area Achievement Standard

By the end of Year 8, students explain how social, ethical, technical and sustainability considerations influence the design of innovative and enterprising solutions to meet a range of present and future needs. They explain how the features of technologies influence design and production decisions. Students make choices between different types of networks for defined purposes.

Students explain a range of needs, opportunities or problems and define them in terms of functional requirements and constraints. They collect, authenticate and interpret data from a range of sources to assist in making informed judgements. Students generate and document in digital and non-digital form, design ideas for different audiences using appropriate technical terms, and graphical representation techniques including algorithms. They independently and safely plan, design, test, modify and create a range of digital solutions that meet intended purposes including user interfaces and the use of a programming language. They plan, document and effectively manage processes and resources to produce designed solutions for each of the prescribed technologies contexts. They develop criteria for success, including innovation and sustainability considerations, and use these to judge the suitability of their ideas, solutions and processes. Students use appropriate protocols when collaborating, and creating and communicating ideas, information and solutions face-to-face and online.

Years 7 and 8 Band Description

Learning in Digital Technologies focuses on further developing understanding and skills in computational thinking such as decomposing problems and prototyping; and engaging students with a wider range of information systems as they broaden their experiences and involvement in national, regional and global activities.

By the end of Year 8, students will have had opportunities to create a range of digital solutions, such as interactive web applications or programmable multimedia assets or simulations of relationships between objects in the real world.

In Year 7 and 8, students analyse the properties of networked systems and their suitability and use for the transmission of data types. They acquire, analyse, validate and evaluate various types of data, and appreciate the complexities of storing and transmitting that data in digital systems. Students use structured data to model objects and events that shape the communities they actively engage with. They further develop their understanding of the vital role that data plays in their lives, and how the data and related systems define and are limited by technical, environmental, economic and social constraints.

They further develop abstractions by identifying common elements while decomposing apparently different problems and systems to define requirements, and recognise that abstractions hide irrelevant details for particular purposes. When defining problems, students identify the key elements of the problems and the factors and constraints at play. They design increasingly complex algorithms that allow data to be manipulated automatically, and explore different ways of showing the relationship between data elements to help computation, such as using pivot tables, graphs and clearly defined mark-up or rules. They progress from designing the user interface to considering user experience factors such as user expertise, accessibility and usability requirements.

They broaden their programming experiences to include general-purpose programming languages, and incorporate subprograms into their solutions. They predict and evaluate their developed and existing solutions, considering time, tasks, data and the safe and sustainable use of information systems, and anticipate any risks associated with the use or adoption of such systems.

Students plan and manage individual and team projects with some autonomy. They consider ways of managing the exchange of ideas, tasks and files, and techniques for monitoring progress and feedback. When communicating and collaborating online, students develop an understanding of different social contexts, for example acknowledging cultural practices and meeting legal obligations.