Scope and sequence

Digital Technologies

Reception to year 6

V1.0
September 2020

Government of South Australia
Department for Education

Digital Technologies: Scope and sequence reception to year 6

Contents

Context statement

Achievement standards

Scope and sequence

- Strand: Knowledge and understanding
  - Thread: Digital systems
  - Thread: Representation of data
- Strand: Process and production skills; creating digital solutions by:
  - Thread: Collecting, managing and analysing data
  - Thread: Investigating and defining
  - Thread: Generating and designing
  - Thread: Producing and implementing
  - Thread: Evaluating
  - Thread: Collaboration and management

Context statement

Digital Technologies is about exciting interest in computer science. It is about empowering students to understand how the devices they use actually work. Students will develop the confidence to create digital solutions.

Computational, systems and design thinking are fundamental to Digital Technologies learning.

Computational thinking is a problem-solving process involving:

- Pattern Recognition: understanding trends, similarities and patterns in data to define problems
- Decomposition: breaking down complex problems into simpler parts
- Abstraction: identifying and removing unnecessary details to simplify a problem
- Algorithmic design: creating a step by step solution to a problem
- Modelling and simulation: implementing the steps to identify and fix 'bugs' or mistakes
- Evaluating: testing out solutions with different audiences. This makes sure the solutions meet the needs of the end user, as well as contribute to preferred futures

Systems thinking is the ability to see the big picture. Students need to understand the impact of digital solutions. They need to see the solution through legal, ethical and sustainability lenses.

Design thinking is the process of imagining, creating and realising solutions.

When using these thinking strategies together, students can develop powerful digital solutions.

This document:

- provides explicit plain English interpretation of the Australian Curriculum content descriptors
- identifies the specific knowledge, skills and understanding learners need at each year level
- guides educators to teach and model computational, systems and design thinking
- supports educators with the processes to design, create and produce digital solutions

Achievement standards

Reception to year 2	Years 3 to 4	Years 5 to 6
By the end of year 2, students: • identify how common digital systems (hardware and software) are used to meet specific purposes • use digital systems to represent simple patterns in data in different ways • design solutions to simple problems using a sequence of steps and decisions • collect familiar data and display them to convey meaning • create and organise ideas and information using information systems, and share information in safe online environments.	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 • explain how the same data sets can be represented in different ways • define simple problems, design and implement digital solutions using algorithms that involve decision-making and user input • explain how the solutions meet their purposes • collect and manipulate different data when creating information and digital solutions • safely use and manage information systems for identified needs using agreed protocols and describe how information systems are used.	By the end of year 6, students: • explain the fundamentals of digital system components (hardware, software and networks) and how digital systems are connected to networks • explain how digital systems use whole numbers as a basis for representing a variety of data types • define problems in terms of data and functional requirements and design solutions by developing algorithms to address the problems • incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program • explain how information systems and their solutions meet needs and consider sustainability • manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols.

Strand: Knowledge and understanding

Threads	Reception	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Digital systems How do computers work and interact? Students use systems thinking to understand how the component parts inside digital systems (like hardware and software) work, both individually and together across networks.	A system is a group of parts that work together. Digital systems include computers, phones, tablets, and other digital devices.	Digital systems have many parts. They work with inputs and outputs. For example • identify parts of digital systems as inputs and outputs – the keyboard and mouse put information in. The screen and speakers send information out.	Digital systems are made up of hardware and software that can be used for specific purposes. For example • hardware is the equipment we can see, touch and feel inside and out of a computer. • software are the programs we use to give instructions to computers. We cannot really touch and feel them but we can see and hear the outputs.	Peripheral devices can be added to a digital system so that it works with more inputs and outputs. For example • Peripheral devices include hardware items like microphones, earphones, cameras and printers.	Digital systems and peripheral devices are used for different purposes and can store and transmit different types of data For example • An image that is input from a camera is broken down into pixels and stored as a smaller file. This image can later be used as an output such as an image in a visual presentation.	Digital systems are made up of many component parts that support the processing of inputs and outputs. The output of a digital system can be images, sound and text that can be sent to other systems in a network. For example • Component parts are the individual pieces of hardware that are part of a larger digital system. • A network is a collection of digital systems that are connected together to allow data to be shared.	Digital systems are made up of many component parts which: • receive inputs • process data into information • store information to be retrieved and used later • produce an output that may be images, sound or text. When digital systems are connected together by wires or wirelessly, they form a network. Networks of digital systems allow data to be shared quickly and efficiently locally and across the world. For example • Artificial Intelligence systems learn from the data it collects and adapts to new situations.

Representation of data How do computers process information? Students use systems thinking to understand data as codes. In upper primary, students build their knowledge so that computers can be understood as integrated systems which deliver images, sound and text using electrical impulses. These impulses are represented using the binary number system.	Symbols and pictures can be used to represent data. Data can include numbers, images, sound and text. For example • universal symbols convey meaning such as - Guiding a friend on a treasure hunt using a map - drawing arrows to represent directions.	Digital systems send and receive information. Information can be represented using symbols. For example • a barrier game where a person describes a simple image while the other person draws is an example of sending and receiving information • identifying the meaning behind the colours on the Aboriginal flag is an example of exploring the ways symbols can represent important ideas.	Patterns in data can be recognised, explored and represented using digital systems. For example • Patterns are repeated sequences such as doubling numbers • Predictions can be made by noticing how a pattern continues.	Agreed ways of representing data helps users to work with digital systems. For example • Agreed codes can be learned and used to send and receive messages • Semaphore and sign language are just two examples of agreed codes.	The same types of data can be represented in different ways including images, sound and text. For example • Bar codes and QR codes are examples of using image codes to indicate information • Creating and solving codes using the practices of cryptography	Images, sound and text can be represented using combinations of whole numbers. For example • Text can be represented as whole numbers using the ASCII (American Standard Code for Information Interchange); images can be represented as whole numbers using the RGB (red, green and blue) values in pixels. • Whole numbers can be represented as binary digits or 1s and 0s.	Whole numbers are used to represent all data in digital systems. The binary number system is a commonly used way of representing data in digital systems using sequences of 1s and 0s. For example • Electrical wires inside digital systems, can only register two states like on or off. Just one wire allows two choices like yes or no, true or false, high or low. This on/off state can be represented with binary digits or bits: 1s and 0s. • Digital systems contain complex electric circuitry, with multiple wires. This allows for many more choices, so that large amounts of data can be represented.

Strand: Process and production skills - Creating digital solutions by:

Threads	Reception	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
Collecting, managing and analysing data How do I find problems that need solving? Students develop the computational thinking strategy of pattern recognition. It is about noticing the patterns in the data that is collected and identifying a problem. They organise data into information so that patterns can be easily seen and problems identified. Students can generate their own data or they can access large 'open data sets' available online. Data is all around us and there are many ways to use data to find patterns. .	Collect, organise and display familiar data using symbols and pictures. Use information from data to pose exploratory questions. For example: • Pose questions such as 'How do we get to school?' With support make statements based on graphically organised data like 'Most people drive to school'	Collect, organise and display familiar data to find patterns in information. Pose questions and make inferences using patterns in data. For example • develop inquiry questions eg 'what do people eat for lunch?' • with support, create graphs so that inferences can be made easily.	Collect, explore and sort familiar data, and use digital systems to present the data creatively. Use information from data investigations to define a need, opportunity or problem that could be solved. For example • make predictions about the answers to a question, eg 'what is a family?' • collect data in response to the question and compare actual results with predictions.	Collect and present different types of data using digital tools to show useful information. Use data sets and investigate information to find problems that need solving. For example • present information from data creatively in order to pinpoint problems, trends and patterns	Collect, access and present different types of data using simple software to create information and solve problems. Define a problem using the needs and opportunities revealed in the information taken from data sets. For example • collect and organise data sets using software, like spreadsheets, to more easily identify a need or problem.	Use a range of software to collect, organise and present new and existing data sets as useful information. For example • investigate questions that are relevant to students' experiences like eg "What is a healthy amount of screen time?" • collect and organise data visually in order to easily identify needs, opportunities or problems related to screen use.	Acquire, store and validate different types of data and use a range of software to interpret and visualise data to create information. For example • investigate data for bias, sources and reliability. • investigate data sets to answer questions that relate to digital solutions eg 'how do we know when our indoor plants are thirsty?' and 'how much exercise should we be doing every day?'
Investigating and defining How do I break a problem down to make it more manageable? Students use the Computational Thinking strategy of decomposition. They decompose (break down) a problem into more manageable parts to make it easier to solve in a logical sequence of steps.	Generate solution ideas for problems identified in data or questions. Break down solution ideas into achievable steps. For example: • consider a global instruction like 'brush your teeth' and break it down into very specific steps.	Generate and organise solution ideas. Identify a preferred solution and break it down into achievable steps. For example • generate the steps to achieve an everyday task like blowing up a balloon or making a sandwich • act out the steps to see if there are any details missing.	Break identified problems or tasks down into workable parts so that a solution can be achieved. For example • represent the steps to a solution in different ways – verbally, a flow chart or diagram. • check to see if the steps are specific enough and can be followed by another user.	Break identified problems down into their component parts and generate solution ideas. Identify digital tools that could be used as part of a solution process. For example • investigate instructions given to 'robots' to arrive at a destination • Make decisions about the best path for it to follow.	Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them. For example • a simple problem may include building a model out of blocks without instructions • identify and record the specific steps to create the model.	Define a problem using data sets as evidence for needs or opportunities. For example • develop a problem statement based on research like 'kids can get hurt if they put too much information online' • investigate solution ideas like providing kid-friendly information about protecting themselves in online situations.	Define problems in terms of data and functional requirements, drawing on previously solved problems. For example • identify problems and define them specifically by breaking them down into solvable parts, eg indoor plants often die because they are watered too much or too little. How can we build a moisture indicator to determine when the plant needs water?
Generating and designing What is the best digital solution to a problem? Students use the Computational Thinking strategy of algorithmic design and abstraction. An algorithm is an efficient and specific sequence of steps to solve a problem. Student develop a step by step process (algorithm) to solve a	Explore simple digital tools and notice how they can be used to solve simple problems. For example • identify the arrow keys on a 'robot' as a simple way to give it instructions	Explore and evaluate simple digital tools based on personal preferences. For example • identify digital tools used by friends and family and how they are used – phone to call a relative, a laptop to write.	Investigate simple digital solutions and how they solve problems for the user. For example • make decisions about which tool to use and for what purpose, using paint software to select and make pictures or using stylus (or finger) to create original drawings.	Identify problems that are solved by digital solutions and suggest how they may change in the future. Identify simple problems and represent the solution as a sequence of steps (algorithm). For example • explore the use of 'robots' in industries • experiment with	Investigate the effectiveness of a range of digital solutions according to the problems they solve or tasks they complete. Design or modify an algorithm that represents a digital solution. Consider ways of showing a decision between 2 or more options For example	Identify the useful features of a digital system and how it solves problems for users. Suggest improvements or modifications. Design or modify an algorithm showing a logical sequence of steps. For example • identify how digital systems meet the needs of diverse users eg screen	Design a user interface for a digital system to solve a problem or perform a particular task. Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition). For example • design a flow chart to demonstrate decisions between two options

problem. They learn how to get rid of unnecessary detail (abstraction) to identify a more efficient set of steps.				using a visual program to instruct a 'robot'.	• develop a flow chart that shows instructions for a simple task with a decision between two options.	magnifiers for a vision impaired person or icons instead of words for a really young user. • develop algorithms that use conditional statements, eg if there is an object in the way, then move around.	(branching) eg yes or no; right or left • design an interactive game or tool that responds to input from a user or sensor.
Producing and implementing What are the steps to solving a problem? Students use the Computational Thinking strategy of modelling and simulation. They try out the steps to solve a problem in a controlled environment. This helps to identify "bugs" in the algorithms	Identify and follow a sequence of steps (algorithm) needed to perform simple tasks or solve a simple problem. For example • use images or verbally describe a sequence such as using a story map to show the beginning, middle and end of a story	Describe and record the steps (algorithm) required to complete a simple task or solve a simple problem, using symbols, picture, diagrams or movements. For example • use images or verbal instructions to describe the steps to achieve a simple task such as making a sandwich	Follow, describe and represent a sequence of steps and decisions (algorithms) needed to solve simple problems or complete simple tasks. For example • arrange images to show how to choose what to wear in different circumstances, eg if it's cold, wear a jumper. If not, wear a t-shirt.	Use a visual program to create a digital solution to a simple problem and test it out on an identified user. For example • observe the way a 'robot' works • design a logical sequence of steps for a 'robot' to follow	Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input For example • create a visual program to direct a character/'robot' through a maze with many dead ends	Identify how a digital solution can be made more efficient by grouping instructions or iterations (repetition). For example • implement a program that uses 'loops' or repeated instructions, for example 'move forward 2 steps until touching a boundary'. • modify a program to refine it into a more efficient set of instructions.	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input For example • develop a flow chart (algorithm) to plan and create a visual program. • manipulate fixed and variable data in a visual program, eg calculate the number of steps taken as measured by a digital pedometer – the 'number' of steps actually taken is the variable data. The goal of 10,000 steps could be the fixed data.

Evaluating Have I created the best solution for the end user? Students use the Computational Thinking strategy of testing. They test out solutions with identified users and seek feedback to improve, further debug or redefine the problem.	Use personal preferences to evaluate the success of simple solutions or the outcomes of simple tasks. For example • play games where answers are yes or no – is it an animal? Is it yellow? • take turns to see who can ask the least questions in order to guess the item	Use agreed criteria to evaluate the success of simple solutions or the outcomes of tasks For example • Consider whether they have used the smallest number of steps to get a 'robot' from one place to another.	Work with others to explore how people safely use common information systems to meet information, communication and recreation needs Develop criteria for success based on user preference. Evaluate solutions against success criteria. For example • interview friends and family about games they enjoy and why • from this, co-develop a criteria rubric for a 'good' game	Evaluate how people use common information and digital solutions to meet common personal, school or community needs. For example • describe common information systems like online encyclopaedias and how they are used to find information. • Identify how to verify an information source as credible.	Explain how student solutions and existing information systems meet common personal, school or community needs. For example • explain information systems used by people in local or global communities and the problems they solve, eg how using mapping software allows us to see places that are far away.	Explain how testing a solution as well as user feedback contribute to the development of common information systems and student solutions. For example • identify errors in existing solutions and adapt or improve upon them (debug) • incorporate user feedback into digital solutions.	Explain how student solutions and existing information systems are sustainable and meet current and future local community needs For example • explain how digital and information systems need updating • test existing solutions or information systems and improve them by incorporating user feedback.
Collaborating and managing How do I work with others to create a solution? Students develop an understanding of how to use agreed social and ethical protocols when designing solutions and when interacting with others online. They also develop an understanding of how to work productively and safely with others in a face to face environment. They develop the skills to manage a project from start to finish.	Work with others to identify and use agreed processes. Collaboratively solve problems and use digital systems safely. For example • with support, co-develop rules for working safely and cooperatively with shared technology tools	Work with others to create and organise ideas and information using information systems safely. Identify trusted networks with whom to safely share information. For example: • create a picture book where each member of the group contributes to a different part of the story. • share stories in safe, teacher managed, digital environments	Create and organise ideas and information using information systems independently and with others, and share these with known people in safe online environments For example • select information from a variety of sources to create a presentation such as images and text to tell a story using a slide deck. • share presentations in secure online environments.	Identify information that is safe to share with others and information that should only be shared with trusted networks. Establish and use practices that demonstrate cyber safety when participating in online environments For example • Create a fictitious character profile and identify which information would be safe for them to share online and which they would need to keep private. • Identify networks of trusted people to report examples of inappropriate online behaviour.	Plan, create and communicate ideas and information independently and with others. Apply agreed ethical and social protocols. For example • develop and use safe and secure processes for communicating information and ideas online • develop a plan to complete a collaborative project from start to finish using appropriate digital tools	Identify the ways to develop positive digital identities and the ways to protect self and others online. Plan, create and communicate ideas and information independently and with others. Apply agreed ethical and social protocol For example • identify and demonstrate the characteristics of a responsible digital citizen • plan and carry out digital projects in collaboration with others using agreed roles and responsibilities	Plan, create and communicate ideas and information, including online collaboration. Apply agreed ethical, social and technical protocols. For example • understand the ways that personal data is collected and used in information systems and digital solutions • establish agreed protocols when creating, managing and producing information in collaborative digital projects.