Structure of the units of work

Every unit of work in the Teach Computing Curriculum contains: a unit overview; a learning graph, to show the progression of skills and concepts in a unit; lesson content — including a detailed lesson plan, slides for learners, and all the resources you will need; and formative and summative assessment opportunities.

Teach Computing Curriculum overview

	Computing systemsand networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 3	Connecting computers (3.1)	Stop-frame animation (3.2)	Sequencing sounds (3.3)	Branching databases (3.4)	Desktop publishing (3.5)	Events and actions in programs (3.6)
Year 4	The internet (4.1)	Audio production (4.2)	Repetition in shapes (4.3)	Data logging (4.4)	Photo editing (4.5)	Repetition in games (4.6)
Year 5	Systems and searching (5.1)	Video production (5.2)	Selection in physical computing (5.3)	Flat-file databases (5.4)	Introduction to vector graphics (5.5)	Selection in quizzes (5.6)
Year 6	Communication and collaboration (6.1)	Webpage creation (6.2)	Variables in games (6.3)	Introduction to spreadsheets (6.4)	3D modelling (6.5)	Sensing movement (6.6)

Unit summaries

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 3	**Connecting computers** Identifying that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks.	**Stop-frame animation** Capturing and editing digital still images to produce a stop-frame animation that tells a story.	**Sequencing sounds** Creating sequences in a block-based programming language to make music.	**Branching databases** Building and using branching databases to group objects using yes/no questions.	**Desktop publishing** Creating documents by modifying text, images, and page layouts for a specified purpose.	**Events and actions in programs** Writing algorithms and programs that use a range of events to trigger sequences of actions.
Year 4	**The internet** Recognising the internet as a network of networks including the WWW, and why we should evaluate online content.	**Audio production** Capturing and editing audio to produce a podcast, ensuring that copyright is considered.	**Repetition in shapes** Using a text-based programming language to explore count-controlled loops when drawing shapes.	**Data logging** Recognising how and why data is collected over time, before using data loggers to carry out an investigation.	**Photo editing** Manipulating digital images, and reflecting on the impact of changes and whether the required purpose is fulfilled.	**Repetition in games** Using a block-based programming language to explore count-controlled and infinite loops when creating a game.

Unit summaries

	Computing systemsand networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 5	**Systems and searching** Recognising IT systems in the world and how some can enable searching on the internet.	**Video production** Planning, capturing, and editing video to produce a short film.	**Selection in physical computing** Exploring conditions and selection using a programmable microcontroller.	**Flat-file databases** Using a database to order data and create charts to answer questions.	**Introduction to vectorgraphics** Creating images in a drawing program by using layers and groups of objects.	**Selection in quizzes** Exploring selection in programming to design and code an interactive quiz.
Year 6	**Communicatio n and collaboration** Exploring how data is transferred by working collaboratively online.	**Webpage creation** Designing and creating webpages, giving consideration to copyright, aesthetics, and navigation.	**Variables in games** Exploring variables when designing and coding a game.	**Introduction to spreadsheets** Answering questions by using spreadsheets to organise and calculate data.	**3D modelling** Planning, developing, and evaluating 3D computer models of physical objects.	**Sensing movement** Designing and coding a project that captures inputs from a physical device.

National curriculum coverage - Years 3 and 4	3.1 Connecting computers	3.2 Stop-frame animation	3.3 Sequencing sounds	3.4 Branching databases	3.5 Desktop publishing	3.6 Events and actions in programs	4.1 The internet	4.2 Audio production	4.3 Repetition in shapes	4.4 Data logging	4.5 Photo editing	4.6 Repetition in games
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts			✓			✓			✓			✓
Use sequence, selection, and repetition in programs; work with variables and various forms of input and output	✓		✓			✓			✓	✓		✓
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs			✓			✓			✓			✓
Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration	✓						✓					
Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content					✓		✓	✓			✓	
Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact		✓		✓			✓	✓			✓	

National curriculum coverage - Years 5 and 6	5.1 Systems and searching	5.2 Video production	5.3 Selection in physical computing	5.4 Flat-file databases	5.5 Introduction to vector graphics	5.6 Selection in quizzes	6.1 Communication and collaboration	6.2 Webpage creation	6.3 Variables in games	6.4 Introduction to spreadsheets	6.5 3D modelling	6.6 Sensing movementz
Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts			✓			✓	✓		✓			✓
Use sequence, selection, and repetition in programs; work with variables and various forms of input and output			✓			✓			✓			✓
Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs			✓			✓			✓			✓
Understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration	✓						✓					
Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content		✓		✓				✓				
Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact	✓	✓						✓	✓		✓	

Progression

Progression across key stages

All learning objectives have been mapped to the National Centre for Computing Education's taxonomy of ten strands, which ensures that units build on each other from one key stage to the next.

Progression across year groups

Within the Teach Computing Curriculum, every year group learns through units within the same four themes, which combine the ten strands of the National Centre for Computing Education's taxonomy (see table, right).

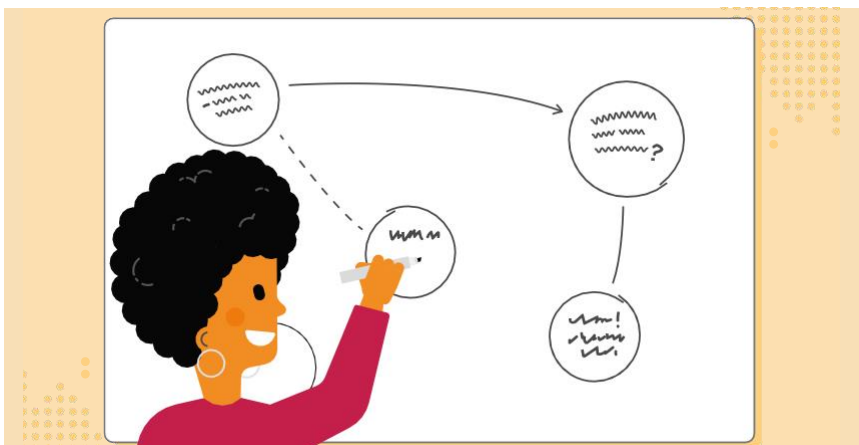
This approach allows us to use the spiral curriculum approach (see the 'Spiral curriculum' section for more information) to progress skills and concepts from one year group to the next.

Primary themes	Computing systems and networks	Programming	Data and information	Creating media
Taxonomy strands	Computer systems Computer networks	Programming Algorithms Design and development	Data and information	Creating media Design and development
	Effective use of tools			
	Impact of technology			
	Safety and security			

Progression within a unit —learning graphs

Learning graphs are provided as part of each unit and demonstrate progression through concepts and skills. In order to learn some of those concepts and skills, pupils need prior knowledge of others, so the learning graphs show which concepts and skills need to be taught first and which could be taught at a different time.

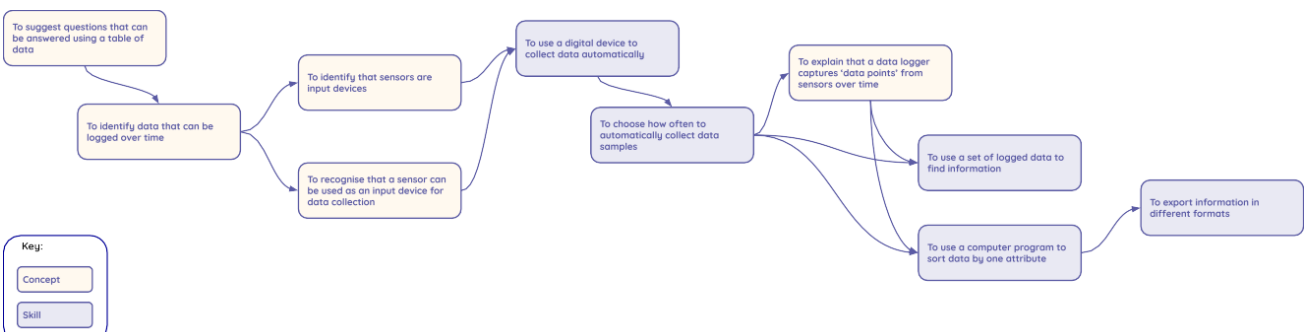
The learning graphs often show more statements than there are learning objectives. All of the skills and concepts learnt are included in the learning graphs. Some of these skills and concepts are milestones, which form learning objectives, while others are smaller steps towards these milestones, which form success criteria. Please note that the wording of the statements may be different in the learning graphs than in the lessons, as the learning graphs are designed for teachers, whereas the learning objectives and success criteria are age-appropriate so that they can be understood by pupils.



In each year group, there are two 'Programming' units of work, but only one 'Programming' learning graph. The second 'Programming' unit builds on the content that was taught in the first 'Programming' unit so closely that there is no specific divide where one ends and the other begins.

KS2 Example learning graph

Year 4 - Data and Information - Data logging



Resources

Software and hardware

Computing is intrinsically linked to technology and therefore requires that pupils experience and use a range of digital tools and devices. As the Teach Computing Curriculum was being written, careful consideration was given to the hardware and software selected for the units. The primary consideration was how we felt a tool would best allow pupils to meet learning objectives; the learning always came first and the tool second.

To make the units of work more accessible to pupils and teachers, the materials include screenshots, videos, and instructions, and these are based on the tools listed in the table below. The list below should not be seen as an explicit requirement for schools. Schools may chooseto use alternative tools that offer the same features as described in the units. All of the learning objectives can be met with alternative hardware and software, as the learning objectives are not designed to be tool-specific.

Software

If you do not wish to use the software recommended in the units, you could use an alternative piece of software that provides the same function. All learning objectives should be achievable using alternative software, however, there will be a lot less support for teachers, as screenshots and demonstration videos reflect the software referenced in the materials.

The units of work include the use of free software that would need to be installed on local machines, and software that is available as an online tool. Where software needs to be installed locally, schools will need to plan software installation in advance.

Several of the units that use online tools require schools to sign up to free services in order to access the tools. This also allows pupils the opportunity to save the projects that they are working on, and gives them the skills that they need to manage their own usernames and passwords as digital citizens. However, the school needs to ensure that they are comfortable using the software, and that it is in line with their policies about using online tools and how teachers will manage accounts.

Hardware

Pupils should experience a range of digital devices, which may include desktop, laptop, and tablet computers. Pupils should also experience hardware designed for specific purposes, e.g. data loggers, floor robots, and microcontrollers.

Several of the Teach Computing Curriculum units require the use of physical computing devices. This is in recognition of the growing importance of physical computing and digital making and was part of our curriculum design from the beginning. As we are aware that not all schools will have invested in this equipment, NCCE Computing Hubs (ncce.io/hubs) have a number of class sets of equipment, which will be loaned to schools in rotation, with some set aside for CPD sessions.

Software and hardware overview

Requirements for pupils — below

	Desktop or laptop	Chromebook	Tablet	Software or hardware
3.1 Connecting computers	✓	●	●	Painting program (any)
3.2 Stop-frame animation	●	●	✓	iMotion (app for iOS)
3.3 Sequencing sounds	✓	✓	●	Scratch
3.4 Branching databases	✓	✓	●	j2data Branch and Pictogram
3.5 Desktop publishing	✓	●	●	Canva.com
3.6 Events and actions in programs	✓	✓	●	Scratch
4.1 The internet	✓	✓	✓	Various websites
4.2 Audio production	✓			Audacity
4.3 Repetition in shapes	✓	●	●	FMSLogo
4.4 Data logging	✓	+	+	Data logger and associated software
4.5 Photo editing	✓	●		Paint.NET (for Microsoft Windows)
4.6 Repetition in games	✓	✓	●	Scratch

✓ Used for the unit — reflected in screenshots ● Could be used as an alternative + Data loggers that work with Chromebooks or tablets are available. Check with suppliers.

Software and hardware overview, cont.

Requirements for pupils — below

	Desktop or laptop	Chromebook	Tablet	Software or hardware
5.1 Systems and Searching	✓	✓		Google Slides
5.2 Video production	✓	●	●	Microsoft Photos (for Microsoft Windows 10)
5.3 Selection in physical computing	✓	✓		Crumble controller + starter kit + motor
5.4 Flat-file databases	✓	✓	●	j2data Database
5.5 Introduction to vector graphics	✓	●		Google Drawings
5.6 Selection in quizzes	✓	✓		Scratch
6.1 Communication and collaboration	✓	✓		Google Slides
6.2 Webpage creation	✓	✓		Google Sites
6.3 Variables in games	✓	✓		Scratch
6.4 Introduction to spreadsheets	✓	✓	●	Google Sheets or Microsoft Excel
6.5 3D modelling	✓	✓	●	Tinkercad
6.6 Sensing movement	✓	✓	●	micro:bit and Microsoft MakeCode

✓ Used for the unit — reflected in screenshots ● Could be used as an alternative