



Progression in computing

Vale Academy Trust

Contents

- Section 1: [Key areas of learning in the primary computing curriculum](#)
- Section 2: [Progression throughout primary school](#)
- Section 3: [Recommended Curriculum and lesson resources](#)
- Section 4: [Recommended hardware and software](#)

Teach Computing Taxonomy

Teachers should be familiar with these different elements of the computing curriculum and there should be ample opportunities to explore these areas throughout primary school.

Teach Computing Taxonomy		
Abbreviation	Strand	Description
NW	Networks	Understand how networks can be used to retrieve and share information, and how they come with associated risks
CM	Creating Media	Select and create a range of media including text, images, sounds, and video
DI	Data & Information	Understand how data is stored, organised, and used to represent real-world artefacts and scenarios
DD	Design & Deveopment	Understand the activities involved in planning, creating, and evaluating computing artefacts
CS	Computing Systems	Understand what a computer is, and how its constituent parts function together as a whole
IT	Impact of Technology	Understand how individuals, systems, and society as a whole interact with computer systems
AL	Algorithms	Be able to comprehend, design, create, and evaluate algorithms
PG	Programming	Create software to allow computers to solve problems
ET	Effective Use of tools	Use software tools to support computing work
SS	Safety & Security	Understand risks when using technology, and how to protect individuals and systems

But what is an algorithm?

- There are lots of technical words in the computing curriculum that can be daunting for teachers. A great guide on the different terms and how they can be taught at different ages (including Early Years) is available at Barefoot computing.
- [Computational thinking concepts and approaches](#)
- [Computer science concepts](#)



Computing at School - Progression in Computing

[This resource](#) is recommended and reviewed by Computing at School. It gives an in-depth overview of progression in different areas in computing from Y1 to Y9.

The purple section indicates the end of primary.

A PDF is also available in the files section.

Computing Progress		
Algorithms	Programming & Development	Data & Data Representation
<ul style="list-style-type: none"> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Understands that computers need precise instructions. (AL) Demonstrates care and precision to avoid errors. (AL) 	<ul style="list-style-type: none"> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text e.g. programmable robots etc. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) 	<ul style="list-style-type: none"> Recognises that digital content can be represented in many forms. (AB) (GE) Distinguishes between some of these forms and can explain the different ways that they communicate information. (AB)
<ul style="list-style-type: none"> Understands that algorithms are implemented on digital devices as programs. (AL) Designs simple algorithms using loops, and selection i.e. if statements. (AL) Uses logical reasoning to predict outcomes. (AL) Detects and corrects errors i.e. debugging, in algorithms. (AL) 	<ul style="list-style-type: none"> Uses arithmetic operators, if statements, and loops, within programs. (AL) Uses logical reasoning to predict the behaviour of programs. (AL) Detects and corrects simple semantic errors i.e. debugging, in programs. (AL) 	<ul style="list-style-type: none"> Recognises different types of data: text, number (AB) (GE) Appreciates that programs can work with different types of data. (GE) Recognises that data can be structured in tables make it useful. (AB) (DE)
<ul style="list-style-type: none"> Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. (AL) Uses diagrams to express solutions. (AB) Uses logical reasoning to predict outputs, showing an awareness of inputs. (AL) 	<ul style="list-style-type: none"> Creates programs that implement algorithms to achieve given goals. (AL) Declares and assigns variables. (AB) Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement. (AL) 	<ul style="list-style-type: none"> Understands the difference between data and information. (AB) Knows why sorting data in a flat file can improve searching for information. (EV) Uses filters or can perform single criteria search for information. (AL)
<ul style="list-style-type: none"> Shows an awareness of tasks best completed by humans or computers. (EV) Designs solutions by decomposing a problem and creates a sub-solution for each of these parts. (DE) (AL) (AB) Recognises that different solutions exist for the same problem. (AL) (AB) 	<ul style="list-style-type: none"> Understands the difference between, and appropriately uses if and if, then and else statements. (AL) Uses a variable and relational operators within a loop to govern termination. (AL) (GE) Designs, writes and debugs modular programs using procedures. (AL) (DE) (AB) (GE) Knows that a procedure can be used to hide the detail with sub-solution. (AL) (DE) (AB) (GE) 	<ul style="list-style-type: none"> Performs more complex searches for information e.g. using Boolean and relational operators. (AL) (GE) (EV) Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions. (AL) (EV)
<ul style="list-style-type: none"> Understands that iteration is the repetition of a process such as a loop. (AL) Recognises that different algorithms exist for the same problem. (AL) (GE) Represents solutions using a structured notation. (AL) (AB) Can identify similarities and differences in situations and can use these to solve problems (pattern recognition). (GE) 	<ul style="list-style-type: none"> Understands that programming bridges the gap between algorithmic solutions and computers. (AB) Has practical experience of a high-level textual language, including using standard libraries when programming. (AB) (AL) Uses a range of operators and expressions e.g. Boolean, and applies them in the context of program control. (AL) Selects the appropriate data types. (AL) (AB) 	<ul style="list-style-type: none"> Knows that digital computers use binary to represent all data. (AB) Understands how bit patterns represent numbers and images. (AB) Knows that computers transfer data in binary. (AL) Understands the relationship between binary and file size (uncompressed). (AB) Defines data types: real numbers and Boolean. (AL) Queries data on one table using a typical query language. (AB)
<ul style="list-style-type: none"> Understands a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. (AL) (GE) Recognises that some problems share the same characteristics and use the same algorithm to solve both. (AL) (GE) Understands the notion of performance 	<ul style="list-style-type: none"> Uses nested selection statements. (AL) Appreciates the need for, and writes, custom functions including use of parameters. (AL) (AB) Knows the difference between, and uses appropriately, procedures and functions. (AL) (AB) 	<ul style="list-style-type: none"> Understands how numbers, images, sounds and character sets use the same bit patterns. (AB) (GE) Performs simple operations using bit patterns e.g. binary addition. (AB) (AL) Understands the relationship between resolution and colour depth, including the effect on file size. (AB)



Sheffield eLearning Primary Computing Progression

- This document utilises the previous document but separates it into year groups and a smaller number of headings. This would be a good place to look to ensure coverage and for assessment purposes.
- [Click here](#) for access to the website. Also available as a PDF in the files section.

Year 3

What is a Computer? Key Skills

- Describe what a computer is (input > process > output).
- Explain the difference between input and output devices on a computer.
- Know where to save and open files (e.g. in shared folder).
- Save files with appropriate names.
- Use a keyboard effectively to type in text.
- Use left-, right- and double-click on the mouse.
- Add an image to a document from the internet.
- Resize and move an image in a document.
- Use a search engine to find simple information.
- Recognise that school computers are connected.

Presenting Information & Multimedia

- Present ideas and information by combining media independently, e.g. text and images.
- Design and create simple digital content for a purpose/audience, e.g. poster.
- Edit digital content to improve it, e.g. resize text.
- Identify the features of a good piece of digital content.
- Explain why we use technology to create digital content.
- Recognise why we use different types of media to convey information, e.g. text, image, audio, video.

Data

- Recognise charts, pictograms and databases, and why we use them.
- Present information using a suitable chart
- Explore a record card database to find out information.
- Use filters in a database to find out specific information.
- Name the key parts of a database, e.g. record, field, search.
- Answer questions about information in a database.
- Name some benefits of using a computer to create charts and databases.
- Recognise that search engines store information in databases.

Programming & Algorithms

- Predict the outcome of a block or text-based program (Scratch/Logo).
- Successfully modify an existing program, e.g. change background, number of times things happen.
- Identify repeated steps in a program or algorithm.
- Create examples of algorithms containing count-controlled loops.
- Use a count-controlled loop (e.g. repeat 3 times) to make a program more efficient.
- Recognise that we can create an algorithm to help plan out a program.
- Recognise a forever loop in a program or algorithm.
- Use a forever loop in a program to keep something happening.
- Identify errors in a block or text-based program and correct them.
- Recognise that different inputs can be used to control a program.

Digital Literacy

- Explain why we need to keep our password safe.
- Recognise that digital content belongs to the person who first created it, but we can give permission for others to use it.
- Recognise when to share personal information and when not to.
- Recognise that some people lie about who they are online.
- Are aware that games and films have age ratings.



Primary Computing Scheme of Work

Teach Computing released a [new free scheme of work](#) in July 2020.

It is very easy for teachers of all confidence levels to access and adapt as necessary.

Teachers will need to create a free login to the site and to download materials.

The website will be regularly updated to ensure resources remain up-to-date.

Most software required is freely available on the web (see list of required resources in the resources section of this ppt) .



Learning Objectives in TC curriculum

In the files section there are Excel spreadsheets of all the lesson objectives for all lessons in one place.

They are linked to the NC and the key learning areas.

There is also an overview of the TC curriculum progression.

NC links for Teach Computing Curriculum KS1



National Curriculum Coverage — Key Stage 1 Computing Curriculum	1.1 Technology around us	1.2 Digital painting	1.3 Moving a robot	1.4 Grouping data	1.5 Digital writing	1.6 Programming animations	2.1 Information technology around us	2.2 Digital photography	2.3 Robot algorithms	2.4 Pictograms	2.5 Making music	2.6 Programming quizzes
Understand what algorithms are, how they are implemented as programs on digital devices, and that programs execute by following precise and unambiguous instructions			✓			✓			✓			✓
Create and debug simple programs			✓			✓			✓			✓
Use logical reasoning to predict the behaviour of simple programs			✓			✓			✓			✓
Use technology purposefully to create, organise, store, manipulate and retrieve digital content	✓	✓		✓	✓	✓	✓	✓		✓	✓	✓
Recognise common uses of information technology beyond school	✓		✓	✓			✓	✓				
Use technology safely and respectfully, keeping personal information private; identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies	✓				✓	✓	✓			✓		

NC links for Teach Computing Curriculum Y3/4



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 editing	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							✓	✓			✓	

NC links for Teach Computing Curriculum Y5/6



National Curriculum Coverage – Years 5 and 6	5.1 Sharing information	5.2 Video editing	5.3 Selection in physical computing	5.4 Flat-file databases	5.5 Vector drawing	5.6 Selection in quizzes	6.1 Internet communication	6.2 Webpage creation	6.3 Variables in games	6.4 Introduction to spreadsheets	6.5 3D modelling	6.6 Sensing
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	✓	✓						✓	✓		✓	

Supplementary lesson plans

[Barefoot Computing](#) have a wealth of lesson plans which concentrate on key computing skills. It has many lessons that don't require any technology to complete. They have a super collection of Scratch resources. Teachers will need to create a free login to download resources.

[Teaching London Computing](#) has lots of resources on computational thinking and problem solving. There are some fun activities that can run alongside other computing topics.

E-safety resources – Foundation, Y1 and Y2

- [Jessie and Friends](#) 3 lessons on watching videos, sharing pictures and playing games (one for Foundation, one for Y1 and one for Y2)
- [Smartie the penguin](#) 2 stories on a similar theme to Jessie and friends. Although they are listed as one per year group on the website, they are all the same! I would recommend doing book 1 in Y1 part way through the year as a refresher of Jessie and Friends and book 2 in Y2 part way through the year as another refresher.
- [Digiduck](#)- more on the same theme. There are 3 stories. I would use Digiduck's big decision in Y1 and Detective Digiduck in Y2. Detective digiduck is all about website reliability so would go well with any topics where you are doing internet research.

E- safety/digital literacy resources KS2

SMART rules intro

- [The Smart Crew – Childnet](#) Introduces the acronym SMART which can be used throughout the keystage. I suggest using this with Y3 – maybe one lesson per term.

Introducing email

- [Password Power up](#) – need to create a free login- (Y4+)
- [Digital etiquette](#) – how to write to others online (including texts)

Think before you click

- [Clickbait](#) – need to create free login – Y5/6 lesson on what clickbait is and how to be aware of it.
- [Don't feed the phish](#) – need to create free login – Y6 identity theft and phishing

Internet reliability

- [All about Explorers](#) (Y4+) This fake website is an amazing resource to see if children can use their digital literacy skills to question the reliability of the website. It has lots of ridiculous information about explorers (also differentiated for reading level) which could be used in a topic separately from the actual teaching on reliability.
- [Trust me](#) - (Y4+) 2 lessons on critical thinking – what should we believe online and what should we click on? Linked to extremism but not in a very obvious way! Can also add into first lesson about 'buying Robux' from scam websites that offer free robux in return for your username etc.

E- safety/digital literacy resources KS2

Images, media and body stereotypes

- [Is seeing believing?](#) – need to create free login – (Y4+) lesson on how images and videos can be faked. Would work well with the image or video lessons from Teach Computing.
- [Lesson on copyright](#) – need to create free login Y5/6
- [Media Smart body image and advertising](#) – couple of lessons on how advertisers can distort body image and how that can impact our own body image. Included video resources and lots of information.
- [Gender stereotypes online](#) – need to create free login – looks at how gender stereotypes can lead to bias or unfairness online
- [Just a joke?](#) - Childnet 3 lessons suitable for Y5/6 on sexualised online bullying (maybe based on gender and identify).

Staying safe online

- [Caught in the web](#) (Y4+) – Newsround special on communicating with and meeting strangers online. Also on spending too much time online – all based on true cases.
- [Is it Cyberbullying?](#) – Y5/6 Looks at cyberbullying and the roles of bystanders and upstanders
- [Keeping games fun and friendly](#) – need to create free login (Y5/6) How to play games online with others
- [Let's fight it together](#) – Y6 video based on real situation of cyberbullying and resolution
- [Live streaming](#) – resources to support this increasingly popular activity for children
- [The biggest threat to you on the internet is yourself](#) – watch the first part of this video with your class. Build on sharing personal information to sharing information such as bad choices, photos, identifying statements.

Hardware needs

Ideally mixture of i-pads/tablets and laptops/chromebooks.

Crumble Starter kits

Could use BBC Microbits or use Crumbles in Y5 and extend in Y6

Digital recording equipment (could be ipads or cameras)

Headphones

Teach Computing Curriculum software and hardware needs - KS1



Software and hardware overview

Requirements for pupils – below

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

	Desktop or laptop	Chromebook	Tablet	Software or hardware
1.1 Technology around us	✓	●	●	paintz.app
1.2 Digital painting	✓	●	●	Microsoft Paint or similar
1.3 Moving a robot				Bee-Bot, Blue-Bot, or other fixed-movement floor robot
1.4 Grouping data	✓	●		Google Slides or Microsoft PowerPoint
1.5 Digital writing	✓	●	●	Google Docs or Microsoft Word
1.6 Programming animations	●	●	✓	ScratchJr
2.1 Information technology around us	✓	●		Google Slides or Microsoft PowerPoint
2.2 Digital photography	✓		●	Digital camera
2.3 Robot algorithms				Bee-Bot, Blue-Bot, or other fixed-movement floor robot
2.4 Pictograms	✓	●	●	j2data Pictogram
2.5 Making music	✓	●	●	Chrome Music Lab
2.6 Programming quizzes	●	●	✓	ScratchJr

Teach Computing Curriculum software and hardware needs - KS2

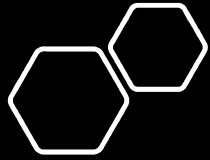
	Desktop or laptop	Chromebook	Tablet	Software or hardware
3.1 Connecting computers	✓	●	●	Painting program (any)
3.2 Stop-frame animation	●	●	✓	iMotion (app for iOS)
3.3 Sequencing sounds	✓	●	●	Scratch
3.4 Branching databases	✓	●	●	j2data Branch and Pictogram
3.5 Desktop publishing	✓	●		Adobe Spark
3.6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

Teach Computing Curriculum software and hardware needs – KS2- cont

	Desktop or laptop	Chromebook	Tablet	Software or hardware
5.1 Sharing information	✓	●		Google Slides
5.2 Video editing	✓	●	●	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 Vector drawing	✓	●		Google Drawings
5.6 Selection in quizzes	✓	●		Scratch
6.1 Internet communication	✓	●		
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	✓	●	●	micro:bit and Microsoft MakeCode

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



NCCE Hub Leader

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 will soon be provided with a number of class sets of equipment, which will be loaned to schools in rotation, with some set aside for CPD sessions.

Our NCCE Hub Leader is Matthew Wimpenny-Smith and his contact details are: TeachComputing@stclementdanes.herts.sch.uk