Digital literacy & IT

To be digitally literate is to be able to **engage the connections and communications possibilities of digital technologies**, in their capacity **to generate, remix, repurpose, and share new knowledge** as well as simply deliver existing information .

Year 3

Pupils have opportunities across the curriculum to select, use and combine a variety of software (including internet services) on a range of digital devices including iPads and laptops to design and create a range content that accomplish given goals, including collecting and presenting data and information.

Year 4

Pupils have opportunities across the curriculum to select, use and combine a variety of software (including internet services) on a range of digital devices including iPads and laptops to design and create a range content that accomplish given goals, including collecting, **analysing**, presenting data and information

Year 5

Pupils have opportunities across the curriculum to select, use and combine a variety of software (including internet services) on a range of digital devices including iPads and laptops to design and create a range content that accomplish given goals, including collecting, analysing, **evaluating** and presenting data and information

Year 6

Pupils have opportunities across the curriculum to select, use and combine a variety of software (including internet services) on a range of digital devices including iPads and laptops to design and create a range content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

	Step 3 (Y3 – Y4)	Step 4 (Y4-Y5)	Step 5 (Y5-Y6)
Online safety	Develop a growing awareness of how to stay safe when using the internet (in school and at home)		Online safety
Research	Using another curriculum area as a starting point, children ask their own questions then use online sources to find answers, making use of search engines, an index, menu, hyperlinks as appropriate. Children use the information or resources they have found.	Make use of copy and paste, beginning to understand the purpose of copyright regulations and the need to repurpose information for a particular audience. They show an understanding that not all information on the internet is accurate.	Independently and with due regard for safety, search the internet using a variety of techniques to find a range of information and resources on a specific topic. Use appropriate methods to validate information and check for bias and accuracy. Repurpose and make appropriate use of selected resources for a given audiences, acknowledging material used where appropriate
Presenting information	Record and present information integrating a range of appropriate media combining text and graphics in printable form and sound and video for on-screen presentations which include hyperlinks. Begin to show an awareness of the intended audience and seek feed-back.	Use advanced tools in word processing / DTP software such as tabs, appropriate text formatting, line spacing etc appropriately to create quality presentations appropriate for a known audience	Multimedia work shows restrained use of effects that help to convey meaning rather than impress.
Digital Images (Video / Still)	Manipulate digital images using a range of tools in appropriate software to convey a specific mood or idea.	Make a short film / animation from images (still and / or moving) that they have sourced, captured or created.	Use images that they have sourced / captured / manipulated as part of a bigger project (eg presentation or document).
Audio	Create a simple audio recording, selecting and importing existing music and sound effects as well as recording their own.	Create their own multiple track compositions that contain a variety of original and pre-recorded sounds.	Create and share more sophisticated audio recordings considering the impact on an audience.
Data (Database)	Children use a simple database (the structure of which has been set up for them) to enter and save and save information on a given subject. They follow straight forward lines of enquiry to search their data for their own purposes. They talk about their experiences of using ICT to process data compared with other methods.	Children work as a class or group to create a data collection sheet and use it to setup a straight forward database to answer questions. Enter information and interrogate it (by searching, sorting, graphing etc). Begin to reflect on how useful the collected data and their interrogation was and whether or not their questions were answered	Independently solve a problem by planning and carrying out data collection, by organising and analysing data involving complex searches using a database, and by drawing conclusions and presenting findings. The need for accuracy is demonstrated and strategies for spotting implausible data are evident. Children able to talk about issues relating to data protection and the need for data security in the world.
Data (Spreadsheets)	Use models and simulations to find things out and solve problems. Recognise that simulations are useful in widening experience beyond the classroom. Make simple use of a spreadsheet to store data and produce graphs	Set up and use a spreadsheet model to explore patterns and relationships. Make predictions. Know how to enter simple formulae to assist this process	Set up and use their own spreadsheet, which contains formulae to investigate mathematical models. Ask "what if …" questions and change variable in their model. Understand the need for accuracy when creating formulae and check regularly for mistakes, by questioning results. Relate their use of spreadsheets to model situations to the wider world.

M Boylan (@ictlinks) November 2019, with thanks to Jane Waite & CAS

Computer Science

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
Appreciate how [search] results are selected and ranked

Computer science & computational thinking allows us to develop skills and techniques to help us solve problems effectively, **with** or **without** the aid of a computer.
Computational thinking is ***not 'thinking like a computer'*** – computers are not capable of thought. Rather, ***it is learning to think in ways which allow us, as humans, to solve problems more effectively*** and, when appropriate, use computers to help us do so.

Year 3	
Step 2	Step 3
Year 4	
Step 3	
Year 5	
Step 4	
Year 6	
Step 4	Step 5

Step2	Step 3	Step 4	Step 5
Use technology purposefully to organise digital content	Use technology responsibly	Design programs that accomplish specific goals	Solve problems by decomposing them into smaller parts
Understand that algorithms are implemented as programs on digital devices	Identify a range of ways to report concerns about contact	Design and create programs	Use selection in programs Work with variables
Understand that programs execute by following precise and unambiguous instructions	Write programs that accomplish specific goals	Debug programs that accomplish specific goals	Use logical reasoning to explain how some simple algorithms work
Debug simple programs	Use sequence in programs	Use repetition in programs	Use logical reasoning to detect and correct errors in algorithms
Use logical reasoning to predict the behaviour of simple programs	Work with various forms of input Work with various forms of output	Control or simulate physical systems	Understand computer networks, including the internet
		Use logical reasoning to detect and correct errors in programs	Appreciate how search results are ranked
		Understand how computer networks can provide multiple services, such as the World Wide Web	
		Appreciate how search results are selected	

DIGITAL TECHNOLOGY
background

Abstraction
Logical thinking
Algorithmic Thinking
Pattern Identification
Decomposition
Evaluation

Computational Thinking = Critical thinking + Power of computing

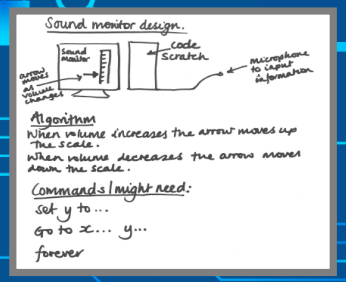
Use computational thinking to analyse the problem and design a solution, including creating an algorithm

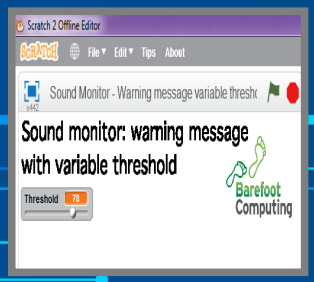
+

Implement these ideas in a programming language on a computer: coding.

=

Programming





Approaches

Copy code	Targeted tasks	Shared coding	Guided exploration	Project design and code	Tinker

- Imitate
- Innovate
- Invent
- Vs
- Remix

PRIMM stands for the following:
- Predict
- Run
- Investigate
- Modify
- Make

Levels of abstraction
Task
Design (including algorithms)
Code
Running the code

M Boylan (@ictlinks) November 2019, with thanks to Jane Waite & CAS

Progression with Sequence

Switch background + no sprites

Single sprites - no concurrency & no coordination

Multiple sprites - concurrency & no coordination

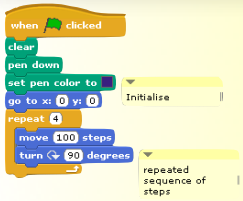
Multiple sprites - concurrency - use time to manage coordinations

Multiple sprites - concurrency - use broadcast to manage coordinations

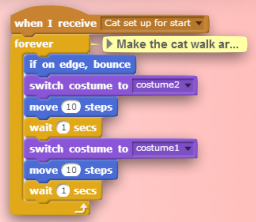
Multiple sprites - concurrency - use broadcast & wait to manage coordinations

Multiple sprites - concurrency - use variables to manage coordinations

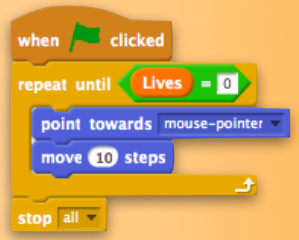
Progression with Repetition



Repeat n times



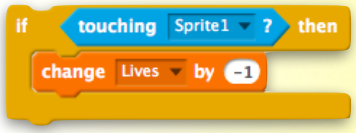
Forever (can be simple or complex - depends on use - spiral progression)



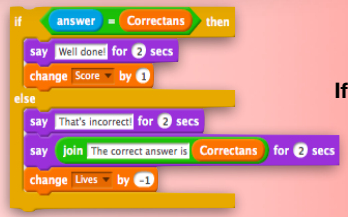
Repeat until (requires boolean and could use input) coordinations

In other programming languages.... for loops / while loop

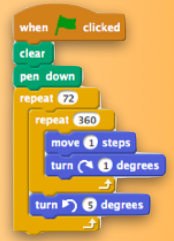
Progression with Selection



If then



If then else



Nested

Event handling/ Forever/ Conditional loops?

Progression with Variables

Use variable to: Display a value that changes during the program e.g. score/lives

Use variable to: Facilitate user to control an aspect of a program e.g. difficulty/ speed/size/colour

Use variable to: Control internal working of the program e.g. difficulty level

Knows what a variable is, can predict what variables will do, can change code with variables, can add a variable....

Progression with own blocks & lists

Make your own block for simple games, for initialisation etc

Suggests using make your own blocks to reuse code, make code readable.

Lists for simple random selection activities

Lists for simple matching activities

For handling data – arrays

M Boylan (@ictlinks) November 2019, with thanks to Jane Waite & CAS, Based on reseeach from Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals, Rich et al 2017