CULCHETH PRIMARY SCHOOL

COMPUTER SCIENCE CURRICULUM

UPDATED 2020

THIS DOCUMENT IS ONLY A GUIDE FOR TEACHERS. OBJECTIVES WILL BE MET BUT ACTIVITIES MAY VARY

Throughout the Computer Science curriculum, children should be taught the following. Vocabulary should be embedded throughout all lessons and taught as appropriate for that age range. Computational Thinking skills are transferable across subjects and are general skills needed to problem solve. The order of teaching Computer Science skills will develop as the children move up through the years. This is indicated accordingly. **N.B. These are only ideas and further ideas can be generated using similar computer science skills as per the age range.**

VOCABULARY	COMPUTATIONAL THINKING SKILLS	COMPUTER SCIENCE SKILLS
Algorithm Code Program Debug Sprite Sequence (order of things) Repetition (repeating to make code more efficient) Selection (asking 'yes / no' questions to make code vary). Logical Reasoning Decomposition Abstraction Variables (data that a computer can remember)	**Decomposition:** breaking down a problem into its parts. **Tinkering:** experimenting and changing things to see what happens. **Abstraction:** identifying what is important and leaving out details we do not need **Debugging:** finding out what is wrong with an algorithm or program and fixing it. **Evaluation:** to make judgements. **Persevering**: never giving up, being determined, resilient and tenacious. **Patterns:** spotting patterns and using these to make predictions, create rules and solve other problems. **Creating:** Planning, making and evaluating things. **Logical thinking:** logic helps us to establish and check facts and also make predictions. **Making alogrithms**: Creating a precise sequence of instructions for performing a task. **Collaborating**: Working together to ensure the best results.	**<u>Foundation, Year 1 and Year 2</u>** Creating algorithms Sequencing Debugging **<u>Year 3 and Year 4</u>** Creating algorithms Sequencing Repetition Debugging Introducing (Variables) **<u>Year 5 and Year 6</u>** Creating algorithms Sequencing Repetition Debugging (Introducing) Variables Selection

<table>
<tr><th colspan="5">YEAR 1</th></tr>
<tr><th>BAREFOOT</th><th>BEE-BOTS</th><th>SWITCHED ON ICT</th><th>iPADS</th><th>NC LINKS</th></tr>
<tr><td>Sharing Sweets Activity
https://barefootcas.org.uk/programme-of-study/understand-algorithms/ks1-sharing-sweets-activity</td><td>Forwards
Backwards
Turning
Making an L Shape</td><td>1.1 We are Treasure Hunters</td><td>Daisy Dinosaur

BeeBot App

A.L.E.X</td><td rowspan="3">• 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</td></tr>
<tr><th>BEBRAS CARDS</th><th colspan="3">SRATCH JNR</th></tr>
<tr><td>Bebras Easy Activities</td><td colspan="3">Can you make the Cat Move?

Can you Make the Sun Set?
https://code.org/files/scratchjr_sunset.pdf</td></tr>
</table>

<table>
<tr><th colspan="3">YEAR 2</th></tr>
<tr><th>BAREFOOT</th><th>BEE-BOTS</th><th>NC LINKS</th></tr>
<tr><td>Beebots Programming 1,2, 3…
https://barefootcas.org.uk/barefoot-primary-computing-resources/concepts/programming/ks1-bee-bots-12-3-programming-activity

Crazy Character Activity
https://barefootcas.org.uk/programme-of-study/understand-algorithms/ks1-crazy-character-algorithms-activity

Pizza Debugging
https://barefootcas.org.uk/programme-of-study/debug-simple-programs/ks1-pizza-pickle-scratch-debugging-activity</td><td>Can the children make the Bee-bot Follow a Path / Map?

BeeBot Activity Cards
http://www.ccc-computing.org.uk/misc/download/zyr5Y9Cgt9e7C9fE

SWITCHED ON ICT

2.1 We are Astronauts
2.2. We are game Testers</td><td rowspan="3">• 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</td></tr>
<tr><th>BEBRAS CARDS</th><th>iPADS</th></tr>
<tr><td>Bebras Medium Activities</td><td>Scratch Jnr (Children to create an animated story)
A.L.E.X
LightBot Activities (up to using Program Function).
(http://lightbot.com/hour-of-code.html)</td></tr>
</table>

YEAR 3		
SCRATCH	**BEBRAS CARDS**	**NC LINKS**
Create a conversation https://csfirst.withgoogle.com/c/cs-first/en/an-unusual-discovery/an-unusual-discovery/an-unusual-discovery.html Knock-Knock Joke Nursery Rhyme (Hickory, Dickory, Dock) Cat and Mouse Game Create a Song https://hourofcode.com/scratchmus Food Maze (https://scratch.mit.edu/projects/381363011/) Drawing Activity (https://scratch.mit.edu/projects/381365957/) Animate Google Logo https://csfirst.withgoogle.com/c/cs-first/en/create-your-own-google-logo/create-your-own-google-logo/create-your-own-google-logo.html Rock Band (https://projects.raspberrypi.org/en/projects/rock-band)	Bebras Medium / Hard Activities **iPADS** LightBot Activities	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour.

YEAR 4		
SCRATCH	**Switched On ICT**	**NC LINKS**
Space Objects Animation (https://scratch.mit.edu/projects/382401861/) Walking Animation (https://scratch.mit.edu/projects/382402065/) Dancing Robot https://scratch.mit.edu/projects/140280486 Frere Jacque Song (https://scratch.mit.edu/projects/382394931/) Quizes (see Switched on ICT). Any similar activities based on Repetition Scratch Studio (https://scratch.mit.edu/studios/2980544/) Simple Voting App (https://scratch.mit.edu/projects/382701604/)	4.1 We are Software Developers **Bee-Bots** Use of Bee-Bots Cars to create shapes (use of angles and also repeat function)	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour.

YEAR 5		
SCRATCH	**Switched On ICT**	**NC LINKS**
Catch Game (incl. score variable) https://scratch.mit.edu/projects/382707482/ Pong Game https://scratch.mit.edu/projects/editor/?tip_bar=pong Tell a Story (https://scratch.mit.edu/projects/382704855/) Quizes based on example (https://scratch.mit.edu/projects/383400455/)	5.1 We are Game Developers **Sphero** Can you make the Sphero Follow a course? Can you get the Sphero to write your name? Block 1 https://edu.sphero.com/cwists/preview/1671x Block 2 https://edu.sphero.com/cwists/preview/2143x Block 3 https://edu.sphero.com/cwists/preview/2152x	• 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 • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour.
TYNKER Carry out similar units to previous challenging the children to use Tyhnker App instead of Scratch (Tynker is similar to Scratch but the children will need to apply their knowledge).		

YEAR 6

SCRATCH

Let's Dance Module
https://scratch.mit.edu/projects/editor/?tip_bar=dance

Frogga Game (create their own)
https://scratch.mit.edu/projects/37144758/#editor

Flappy Bird Game
(https://scratch.mit.edu/projects/383410377/)

Classification keys (based on
https://scratch.mit.edu/projects/383406551/)

SWIFT PLAYGROUNDS

https://www.apple.com/uk/swift/playgrounds/

Children to complete the Swift Playgrounds *lessons*. This is to give the children an experience of word based coding rather than block coding.

Sphero

Sphero Colour Challenge (navigate course and when it lands of a colour square, sphero should change colour). Can children Code this using Java Script?
https://edu.sphero.com/cwists/preview/110x

Block 4
https://edu.sphero.com/cwists/preview/6933x

Sphero Hopscotch
https://edu.sphero.com/cwists/preview/383x

Sphero Area of Rectangle
https://edu.sphero.com/cwists/preview/15583x

NC LINKS

- 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
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour.

N.B. **TYNKER** Carry out similar units to previous challenging the children to use Tynker App instead of Scratch (Tynker is similar to Scratch but the children will need to apply their knowledge).