

# St Edward's Computing Curriculum Map 2020-2021

	Coding and Computational thinking		Spreadsheets		Internet and Email		Art and Design		Music		Databases and graphing		Writing and Presenting		Communication and networks
--	-----------------------------------	--	--------------	--	--------------------	--	----------------	--	-------	--	------------------------	--	------------------------	--	----------------------------

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
YEAR 1	Unit 1.1 Online Safety & Exploring Purple Mash				Unit 1.2 Grouping & Sorting		Unit 1.3 Pictograms		Unit 1.4 Lego Builders			Unit 1.5 Maze Explorers			Unit 1.6 Animated Story Books			Unit 1.7 Coding			Unit 1.8 Spreadsheets		Unit 1.9 Technology outside school								
	Weeks – 4				Weeks – 2		Weeks – 3		Weeks – 3			Weeks – 3			Weeks – 5			Weeks – 6			Weeks – 3		Weeks – 2								
	Programs – Various				Programs – 2DIY		Programs – 2Count		Programs – 2DIY			Programs – 2Go			Programs – 2Create A Story			Programs – 2Code			Programs – 2Calculate		Programs – Various								

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
YEAR 2	Unit 2.1 Coding					Unit 2.2 Online Safety			Unit 2.3 Spreadsheets				Unit 2.4 Questioning				Unit 2.5 Effective Searching		Unit 2.6 Creating Pictures			Unit 2.7 Making Music		Unit 2.8 Presenting Ideas								
	Weeks – 5					Weeks – 3			Weeks – 4				Weeks – 5				Weeks – 3		Weeks – 5			Weeks – 3		Weeks – 4								
	Programs – 2Code					Programs – Various			Programs – 2Calculate				Programs – 2Question, 2Investigate				Programs – Browser		Programs – 2PaintAPicture			Programs – 2Sequence		Programs – Various								



As of September 2020 all year groups will be following the PurpleMash scheme of computing. For this year only we may run slightly behind as children are also being taught to use Office 365 in the case of a lockdown due to Covid19.

# St Edward's Computing Curriculum Map 2020-2021

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
YEAR 3	Unit 3.1 Coding						Unit 3.2 Online safety			Unit 3.3 Spreadsheets			Unit 3.4 Touch Typing			Unit 3.5 Email (including email safety)				Unit 3.6 Branching Databases		Unit 3.7 Simulations		Unit 3.8 Graphing									
	Number of Weeks – 6						Weeks – 3			Weeks – 3			Weeks – 4			Weeks – 6				Weeks – 4		Weeks – 3		Weeks – 3									
	Main Programs – 2Code						Programs – Various			Programs – 2Calculate			Programs – 2Type			Programs – 2Email, 2Connect, 2DIY				Programs – 2Question		Programs – 2Simulate, 2Publish		Programs – 2Graph									
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
YEAR 4	Unit 4.1 Coding						Unit 4.2 Online safety			Unit 4.3 Spreadsheets				Unit 4.4 Writing for different audiences				Unit 4.5 Logo		Unit 4.6 Animation		Unit 4.7 Effective Search		Unit 4.8 Hardware Investigators									
	Number of Weeks – 6						Weeks – 4			Weeks – 6				Weeks – 5				Weeks – 4		Weeks – 3		Weeks – 3		Weeks – 2									
	Main Programs – 2Code						Programs – Various			Programs – 2Calculate				Programs – 2Email, 2Connect, 2DIY				Programs – Logo		Programs – 2Animate		Programs – Browser											

**Unit 3.9  
Presenting (with  
Microsoft PowerPoint)**  
  
(Optional Unit)  
**Number of Lessons – 5  
or 6 (version dependent)**  
**Main Program – MS  
PowerPoint**

**Unit 4.9  
Making Music**  
  
(Optional Unit)  
**Number of Lessons – 4**  
  
**Main Program – Busy  
Beats**

# St Edward's Computing Curriculum Map 2020-2021

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
YEAR 5	<b>Unit 5.1</b> Coding						<b>Unit 5.2</b> Online safety			<b>Unit 5.3</b> Spreadsheets						<b>Unit 5.4</b> Databases				<b>Unit 5.5</b> Game Creator				<b>Unit 5.6</b> 3D Modelling			<b>Unit 5.7</b> Concept Maps					
	Number of Weeks – 6						Weeks – 3			Weeks – 6						Weeks – 4				Weeks – 5				Weeks – 4			Weeks – 4					
	Main Programs – 2Code						Programs - Various			Programs – 2Calculate						Programs – 2Question, 2Investigate				Programs – 2DIY 3D				Programs – 2Design and Make			Programs – 2Connect					

**Unit 5.8**

**Word processing (with Microsoft Word)**

(Optional Unit)

**Number of Lessons – 8**

**Main program – MS Word**

# St Edward's Computing Curriculum Map 2020-2021

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
YEAR 5	<b>Unit 5.1</b> Coding						<b>Unit 5.2</b> Online safety			<b>Unit 5.3</b> Spreadsheets						<b>Unit 5.4</b> Databases			<b>Unit 5.5</b> Game Creator			<b>Unit 5.6</b> 3D Modelling		<b>Unit 5.7</b> Concept Maps								
	Number of Weeks – 6						Weeks – 3			Weeks – 6						Weeks – 4			Weeks – 5			Weeks – 4		Weeks – 4								
	Main Programs – 2Code						Programs - Various			Programs – 2Calculate						Programs – 2Question, 2Investigate			Programs – 2DIY 3D			Programs – 2Design and Make		Programs – 2Connect								

**Unit 5.8**

**Word processing (with Microsoft Word)**

(Optional Unit)

**Number of Lessons – 8**

**Main program – MS Word**

# St Edward's Computing Curriculum Map 2020-2021

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
YEAR 6*	Unit 6.1 Coding						Unit 6.2 Online safety			Unit 6.3 Spreadsheets					Unit 6.4 Blogging				Unit 6.5 Text Adventures				Unit 6.6 Networks			Unit 6.7 Quizzing						
	Number of Weeks – 6						Weeks – 2			Weeks – 5					Weeks – 5				Weeks – 5				Weeks – 3			Weeks – 6						
	Main Programs – 2Code						Programs - Various			Programs – 2Calculate					Programs – 2Blog				Programs – 2Code, 2Connect							Programs – 2Quiz, 2DIY, Text Toolkit, 2Investigate						

**Unit 6.8**  
**Understanding Binary**  
 (Optional Unit)  
 Number of Lessons – 4  
 Main Program – 2Code

**Unit 6.9**  
**Spreadsheets (with Microsoft Excel)**  
 (Optional Unit)  
 Number of Lessons – 8  
 Main program – MS Excel

## Tools by Unit

Year	Unit	Title	Tools used
Y1	1.1	Online Safety and Exploring Purple Mash	Avatar creator
			Paint Projects
			Writing Templates
			2Count (Pictograms)
			2Explore (Music)
	1.2	Grouping & Sorting	2Quiz
	1.3	Pictograms	2Connect (Mind Map)
			2Count (Pictograms)
	1.4	Lego Builders	Paint Projects
Writing Templates			
2Quiz			
1.5	Maze Explorers	2Go (coding)	
1.6	Animated Stories	2Create a Story	
1.7	Coding	2Code	
1.8	Spreadsheets	2Calculate	
1.9	Technology Outside School	Writing Templates	

Year	Unit	Title	Tools used
Y3	3.1	Coding	2Code
	3.2	Online Safety	2Connect (Mind Map)
			2Blog (Blogging)
			Writing Templates
			Displayboards
	3.3	Spreadsheets	2Calculate
	3.4	Typing	2Type
	3.5	Email	2Email
	3.6	Branching Databases	2Question (Binary Databases)
3.7	Simulations	2Simulate	
		Writing Templates	
3.8	Graphing	2Graph	
		Writing Templates	
		2Blog (Blogging)	
3.9	Presenting (with Microsoft PowerPoint)	Microsoft PowerPoint	

Year	Unit	Title	Tools used
Y2	2.1	Coding	2Code
	2.2	Online Safety	Writing Templates
			Displayboards
			2Respond (2Email)
	2.3	Spreadsheets	2Calculate
	2.4	Questioning	2Question (Binary Databases)
			2Calculate (spreadsheet)
			2Investigate (database)
	2.5	Effective Searching	2Quiz
			Writing Templates
	2.6	Creating Pictures	2Paint a Picture
Writing Templates			
2.7	Making Music	2Sequence (Music)	
2.8	Presenting Ideas	2Connect (Mind Map)	
		2Create a Story (ebook)	
		2Quiz	
		Writing Templates	

Year	Unit	Title	Tools used
Y4	4.1	Coding	2Code
	4.2	Online Safety	2Connect (Mind Map)
			2Publish Plus
			Displayboards
	4.3	Spreadsheets	2Calculate
	4.4	Writing for Different Audiences	Writing Templates
			2Simulate
			2Connect (Mind Map)
			2Publish Plus
	4.5	Logo	2Logo (text-based coding)
	4.6	Animation	2Animate
4.7	Effective Searching	2Quiz	
		2Connect (Mind Map)	
4.8	Hardware Investigators	2Quiz	
		2Connect (Mind Map)	
		Writing Templates	
4.9 (Optional)	Making Music	Busy Beats	
		2Sequence	
		Writing Templates	

# St Edward's Computing Curriculum Map 2020-2021

Year	Unit	Title	Tools used
Y5	5.1	Coding	2Code
	5.2	Online Safety	2Publish Plus
			Writing Templates
			Displayboards
			2Connect (Mind Map)
	5.3	Spreadsheets	2Calculate
	5.4	Databases	2Investigate (database)
			Avatar creator
5.5	Game Creator	2DIY 3D	
		Writing Templates	
		2Blog (Bloggng)	
5.6	3D Modelling	2Design and Make	
		Writing Templates	
5.7	Concept Maps	2Connect (Mind Map)	
5.8	Word processing (with Microsoft Word)	MS Word	

Year	Unit	Title	Tools used
Y6	6.1	Coding	2Code
	6.2	Online Safety	2DIY 3D 2DIY 2Code
			2Blog (Bloggng)
	6.3	Spreadsheets	2Calculate
	6.4	Bloggng	2Blog (Bloggng)
	6.5	Text Adventures	2Code
			2Connect (Mind Map)
			Writing Templates
	6.6	Networks	2Connect (Mind Map)
Writing Templates			
6.7	Quizzng	2DIY	
		2Quiz	
		Text Toolkit	
		2Investigate (database)	
6.8 (optional)	Understanding Binary	2Connect (Mind Map)	
		2Question (Binary Databases)	
		Writing Templates	
		2Code	
6.9 (optional)	Spreadsheets (with Microsoft Excel)	MS Excel	



## Computing Progression N.C. Statements KS1 Year 1

	Computer Science			Information Technology	Digital Literacy	
Statement	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.
Outcome	Children understand that an algorithm is a set of instructions used to solve a problem or achieve an objective. They know that an algorithm written for a computer is called a program.	Children can work out what is wrong with a simple algorithm when the steps are out of order, e.g. <a href="#">The Wrong Sandwich</a> in Purple Mash and can write their own simple algorithm, e.g. <a href="#">Colouring in a Bird activity</a> . Children know that an unexpected outcome is due to the code they have created and can make logical attempts to fix the code, e.g. <a href="#">Bubbles activity in 2Code</a> .	When looking at a program, children can read code one line at a time and make good attempts to envision the bigger picture of the overall effect of the program. Children can, for example, interpret where the turtle in <a href="#">2Go challenges</a> will end up at the end of the program.	Children are able to sort, collate, edit and store simple digital content e.g. children can name, save and retrieve their work and follow simple instructions to access online resources, use Purple Mash <a href="#">2Quiz</a> example (sorting shapes), <a href="#">2Code</a> design mode (manipulating backgrounds) or using pictogram software such as <a href="#">2Count</a> .	Children understand what is meant by technology and can identify a variety of examples both in and out of school. They can make a distinction between objects that use modern technology and those that do not e.g. a microwave vs. a chair.	Children understand the importance of keeping information, such as their usernames and passwords, private and actively demonstrate this in lessons. Children take ownership of their work and save this in their own private space such as their My Work folder on Purple Mash.



## Computing Progression N.C. Statements KS1 Year 2



	Computer Science			Information Technology	Digital Literacy	
Statement	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.
Outcome	Children can explain that an algorithm is a set of instructions to complete a task. When <a href="#">designing simple programs</a> , children show an awareness of the need to be precise with their algorithms so that they can be successfully converted into code.	Children can create a simple program that achieves a specific purpose. They can also identify and correct some errors, e.g. <a href="#">Debug Challenges: Chimp</a> . Children's program designs display a growing awareness of the need for logical, programmable steps.	Children can identify the parts of a program that respond to specific events and initiate specific actions. For example, they can write a cause and effect sentence of what will happen in a program.	Children demonstrate an ability to organise data using, for example, a database such as <a href="#">2Investigate</a> and can retrieve specific data for conducting simple searches. Children are able to edit more complex digital data such as music compositions within <a href="#">2Sequence</a> . Children are confident when creating, naming, saving and retrieving content. Children use a range of media in their digital content including photos, text and sound.	Children can effectively retrieve relevant, purposeful digital content using a search engine. They can apply their learning of effective searching beyond the classroom. They can share this knowledge, e.g. <a href="#">2Publish example template</a> . Children make links between technology they see around them, coding and multimedia work they do in school e.g. <a href="#">animations, interactive code</a> and <a href="#">programs</a> .	Children know the implications of inappropriate online searches. Children begin to understand how things are shared electronically such as posting work to the Purple Mash display board. They develop an understanding of using email safely by using <a href="#">2Respond</a> activities on Purple Mash and know ways of reporting inappropriate behaviours and content to a trusted adult.

## Computing Progression N.C. Statements KS2 Year 3



	Computer Science				Information Technology		Digital Literacy
Statement	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 concern about content and contact.
Outcome	Children can turn a simple real-life situation into an algorithm for a program by deconstructing it into manageable parts. Their design shows that they are thinking of the desired task and how this translates into code. Children can identify an error within their program that prevents it following the desired algorithm and then fix it.	Children demonstrate the ability to design and code a program that follows a simple sequence. They experiment with timers to achieve repetition effects in their programs. Children are beginning to understand the difference in the effect of using a timer command rather than a repeat command when creating repetition effects. Children understand how variables can be used to store information while a program is executing.	Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'if' statements, repetition and variables. They make good attempts to 'step through' more complex code in order to identify errors in algorithms and can correct this. e.g. traffic light algorithm in <a href="#">2Code</a> . In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately.	Children can list a range of ways that the internet can be used to provide different methods of communication. They can use some of these methods of communication, e.g. being able to open, respond to and attach files to emails using <a href="#">2Email</a> . They can describe appropriate email conventions when communicating in this way.	Children can carry out simple searches to retrieve digital content. They understand that to do this, they are connecting to the internet and using a search engine such as Purple Mash search or internet-wide search engines.	Children can collect, analyse, evaluate and present data and information using a selection of software, e.g. using a branching database ( <a href="#">2Question</a> ), using software such as <a href="#">2Graph</a> . Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails, e.g. <a href="#">2Respond</a> .	Children demonstrate the importance of having a secure password and not sharing this with anyone else. Furthermore, children can explain the negative implications of failure to keep passwords safe and secure. They understand the importance of staying safe and the importance of their conduct when using familiar communication tools such as <a href="#">2Email</a> in Purple Mash. They know more than one way to report unacceptable content and contact.

## Computing Progression N.C. Statements KS2 Year 4



	Computer Science				Information Technology		Digital Literacy
Statement	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 concern about content and contact.
Outcome	When turning a real-life situation into an algorithm, the children's design shows that they are thinking of the required task and how to accomplish this in code using coding structures for selection and repetition. Children make more intuitive attempts to debug their own programs.	Children's use of timers to achieve repetition effects are becoming more logical and are integrated into their program designs. They understand 'if statements' for selection and attempt to combine these with other coding structures including variables to achieve the effects that they design in their programs. As well as understanding how variables can be used to store information while a program is executing, they are able to use and manipulate the value of variables. Children can make use of user inputs and outputs such as 'print to screen'. e.g. <a href="#">2Code</a> .	Children's designs for their programs show that they are thinking of the structure of a program in logical, achievable steps and absorbing some new knowledge of coding structures. For example, 'if' statements, repetition and variables. They can trace code and use step-through methods to identify errors in code and make logical attempts to correct this. e.g. traffic light algorithm in <a href="#">2Code</a> . In programs such as Logo, they can 'read' programs with several steps and predict the outcome accurately.	Children recognise the main component parts of hardware which allow computers to join and form a network. Their ability to understand the online safety implications associated with the ways the internet can be used to provide different methods of communication is improving.	Children understand the function, features and layout of a search engine. They can appraise selected webpages for credibility and information at a basic level. .	Children are able to make improvements to digital solutions based on feedback. Children make informed software choices when presenting information and data. They create linked content using a range of software such as <a href="#">2Connect</a> and <a href="#">2Publish+</a> . Children share digital content within their community, i.e. using <a href="#">Virtual Display Boards</a> .	Children can explore key concepts relating to online safety using concept mapping such as <a href="#">2Connect</a> . They can help others to understand the importance of online safety. Children know a range of ways of reporting inappropriate content and contact.

## Computing Progression N.C. Statements KS2 Year 5



	Computer Science				Information Technology		Digital Literacy
Statement	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 concern about content and contact.
Outcome	Children may attempt to turn more complex real-life situations into algorithms for a program by deconstructing it into manageable parts. Children are able to test and debug their programs as they go and can use logical methods to identify the approximate cause of any bug but may need some support identifying the specific line of <u>code</u> .	Children can translate algorithms that include sequence, selection and repetition into code with increasing ease and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures. They are combining sequence, selection and repetition with other coding structures to achieve their <u>algorithm design</u> .	When children code, they are beginning to think about their code structure in terms of the ability to debug and interpret the code later, e.g. the use of tabs to organise code and the <u>naming of variables</u> .	Children understand the value of computer networks but are also aware of the main dangers. They recognise what personal information is and can explain how this can be kept safe. Children can select the most appropriate form of online communications contingent on audience and digital content, e.g. <u>2Blog, 2Email, Display Boards</u> .	Children search with greater complexity for digital content when using a search engine. They are able to explain in some detail how credible a webpage is and the information it contains.	Children are able to make appropriate improvements to digital solutions based on feedback received and can confidently comment on the success of the solution. e.g. creating their own program to meet a design brief using <u>2Code</u> . They objectively review solutions from others. Children are able to collaboratively create content and solutions using digital features within software such as collaborative mode. They are able to use several ways of sharing digital content, i.e., <u>2Blog, Display Boards and 2Email</u> .	Children have a secure knowledge of common online safety rules and can apply this by demonstrating the safe and respectful use of a few different technologies and <u>online services</u> . Children implicitly relate appropriate online behaviour to their right to personal privacy and mental wellbeing of themselves and others.



## Computing Progression N.C. Statements KS2 Year 6

	Computer Science				Information Technology		Digital Literacy
Statement	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 concern about content and contact.
Outcome	Children are able to turn a more complex programming task into an algorithm by identifying the important aspects of the task (abstraction) and then decomposing them in a logical way using their knowledge of possible coding structures and applying skills from previous programs. Children test and debug their program as they go and use logical methods to identify the cause of bugs, demonstrating a systematic approach to try to identify a particular line of code causing a <u>problem</u> .	Children translate algorithms that include sequence, selection and repetition into code and their own designs show that they are thinking of how to accomplish the set task in code utilising such structures, including nesting structures within each other. Coding displays an improving understanding of variables in coding, outputs such as sound and movement, inputs from the user of the program such as button clicks and the <u>value of functions</u> .	Children are able to interpret a program in parts and can make logical attempts to put the separate parts of a complex algorithm together to explain the <u>program as a whole</u> .	Children understand and can explain in some depth the difference between the internet and the World Wide Web. Children know what a WAN and LAN are and can describe how they access the <u>internet in school</u> .	Children readily apply filters when searching for digital content. They are able to explain in detail how credible a webpage is and the information it contains. They compare a range of digital content sources and are able to rate them in terms of content quality and accuracy. Children use critical thinking skills in everyday use of online communication.	Children make clear connections to the audience when designing and creating digital content. The children design and create their own blogs to become a content creator on the internet, e.g. <u>2Blog</u> . They are able to use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.	Children demonstrate the safe and respectful use of a range of different technologies and online services. They identify more discreet inappropriate behaviours through developing critical thinking, e.g. <u>2Respond</u> activities. They recognise the value in preserving their privacy when online for their own and other people's safety.