The Whartons Primary School- Computing Progression of Knowledge						
Curriculum Area/ Big Ideas	**Y1**	**Y2**	**Y3**	**Y4**	**Y5**	**Y6**
Vocabulary	**Unit 1.1:** Log in, log out, Username, password, My work, Avatar, Notification, Topics, tools, save. **Unit 1.2:** Sort, criteria **Unit 1.3:** Pictogram, data, collate **Unit 1.4:** Instruction, Algorithm, computer, program, debug **Unit 1.5:** Direction, challenge, arrow, undo, rewind, forward, backwards, right turn, left turn **Unit 1.6:** Animation, e-book, font, file, sound effect, display board **Unit 1.7:** Action, background, button, character, code block, code design, coder, coding, collision detection, command input, design mode, object, scale, properties, stop command, sound, when clicked, when key **Unit 1.8:** Arrow keys, backspace key, cursor, columns, cells, clipart, count tool, delete key, move cell tool, image toolbox, lock tool, move cell tool, rows, spreadsheet, speak tool. **Unit 1.9:** Technology	**Unit 2.1:** Action, algorithm, bug, character, code block, code design, command, debug/debugging, design mode, input, object, properties, repeat, scale, timer, when clicked, when key **Unit 2.2:** Search, display boards, internet, sharing, email, attachment, digital footprint **Unit 2.3:** Backspace key, copy & paste, columns, cells, count tool, delete key, equals tool, image toolbox, lock tool, move cell tool, rows, speak tool, spreadsheet **Unit 2.4:** Pictogram, question, data, collate, binary tree, avatar, database **Unit 2.5:** Internet, search, search engine **Unit 2.6:** Impressionism, palette, pointillism, share, surrealism, template **Unit 2.7:** Bpm, composition, digitally, instrument, music, sound effects (sfx), soundtrack, tempo, volume **Unit 2.8:** Concept map (mind map), quiz, presentation, node, animated, non-fiction, narrative, audience	**Unit 3.1:** Action, algorithm, bug, character, code block, code design, command, debug/debugging, design mode, event, input, object, if, repeat, properties, timer, computer simulation, selection, variable **Unit 3.2:** Password, internet, blog, concept map, username, website, webpage, spoof website, PEGI rating **Unit 3.3:** copy & paste, columns, cells, advance mode, delete key, equals tool, spin tool, move cell tool, rows, spreadsheet **Unit 3.4:** Posture, top row keys, home row keys, bottom row keys, space bar **Unit 3.5:** communication, email, compose, send, cc, attachment, formatting, report to the teacher, password, address book, save to draft **Unit 3.6:** Data, database, branching database, question **Unit 3.7:** Simulation **Unit 3.8:** Graph, field, data, bar chart, block graph, line graph **Unit 3.9:** Animation, audio, design templates, entrance animation, font, media, presentation, presentation program, slide, slideshow, stock image, textbox, text formatting, transition	**Unit 4.1:** Action, alert, algorithm, code design, control, command, debug/debugging, design mode, event, flowchart bug, get input, if, if/else, input, object, repeat, selection, computer simulation, simulation, timer, variable **Unit 4.2:** Computer virus, cookies, copyright, digital footprint, email, identity theft, malware, phishing, plagiarism, spam **Unit 4.3:** Average, advance mode, copy & paste, columns, cells, charts, equals tool, formular, formular wizard, move cell tool, random tool, rows, spin tool, spreadsheet, timer **Unit 4.4:** Font, bold, italic, underline **Unit 4.5:** Logo, BK (backwards), FD (forwards), RT (Right turn), LT (Left turn), repeat, setpc (set pen colour), setps (set pen thickness), PU (pen up), PD (pen down) **Unit 4.6:** Animation, background, frame, flipbook, onion skinning, stop motion, play, sound, video clip **Unit 4.7** Easter egg, internet, internet browser, search, search engine, spoof website, website **Unit 4.8** Motherboard, CPU, RAM, graphics card, network card, monitor, speakers, keyboard, mouse **Unit 4.9** Pitch, rhythm, pulse, tempo, dynamics, melody, rippler, house music, texture	**Unit 5.1:** Action, alert, algorithm, bug, code design, command, control, debug/debugging, design mode, event, get input, if, if/else, input, object, output, repeat, selection, simulation, sequence, timer, variable **Unit 5.2:** Online safety, smart rules, password, reputable, encryption, identity theft, shared image, plagiarism, citations, reference, bibliography **Unit 5.3:** Average, advance mode, copy & paste, columns, cells, charts, equals tool, formular, formular wizard, move cell tool, random tool, rows, spin tool, spreadsheet, timer **Unit 5.4:** Avatar, binary tree (branching database), charts, collaborative, data, database, find, record, sort, group & arrange, statistics & reports, table **Unit 5.5:** Animation, computer game, customise, evaluation, image, instructions, interactive, screenshot, texture, perspective, playability **Unit 5.6:** CAD, modelling, 3D, 2D, viewpoint, polygon, net, 3d printing, points, template **Unit 5.7:** Audience, collaboratively, concept, concept map, connection, idea, node, thought, visual **Unit 5.8:** Copyright, cursor, document, font, in-built styles, merge cells, paragraph formatting, readability, template, text formatting, text wrapping, word art, word processing tool	**Unit 6.1:** Action, alert, algorithm, code design, command, control, debug/debugging, event, flow chart, function, get input, flowchart bug, if/else, input, object, output, repeat, simulation, tabs, selection, sequence, timer, variable **Unit 6.2:** Digital footprint, password, PEGI rating, phishing, screen time, spoof website **Unit 6.3:** Average, advance mode, copy & paste, columns, cells, charts, count tool, dice, equals tool, formular, formular wizard, move cell tool, random tool, rows, spin tool, spreadsheet, timer **Unit 6.4:** Audience, blog, blog page, blog post, collaborative, icon **Unit 6.5:** Text-based adventure, concept map, debug, sprite, function **Unit 6.6:** Internet, WWW, network, router, local area network (LAN), wide area network (WAN), network cables, wireless **Unit 6.7:** Audience, collaboration, concept map, database, quiz **Unit 6.8:** Base 10, Base 2, Binary, Bit, byte, decimal, gigabyte (GB), denary, digit, machine code, integer, kilobyte (KB), switch, megabyte (MB), nibble (4 bits), tetrabyte (TB 1024GB), Transistor, variable **Unit 6.9:** Alignment, calculation, cell, cell reference, chart, column, formula(e), function, range, row, spreadsheet, style, sum, text wrapping, value, workbook

Computer science	- 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 experiment with algorithms. - Children can work out what is wrong with a simple algorithm when the steps are out of order, e.g. The Wrong Sandwich in Purple Mash and can write their own simple algorithm, e.g. Colouring in a Bird activity. 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. Bubbles activity in 2Code. - 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 2Go challenges will end up at the end of the program.	- Children can explain that an algorithm is a set of instructions to complete a task. When designing simple programs, 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. Debug Challenges: Chimp. 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 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 2Code. 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 2Email. They can	- 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. 2Code - 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 stepthrough methods to identify errors in code and make logical attempts to correct this. e.g. traffic light algorithm in 2Code. 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	- 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 code. - 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 algorithm design. - 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 naming of variables. - 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. 2Blog, 2Email, Display Boards.	- 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 problem. - 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 value of functions. - 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 program as a whole. - 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 internet in school.

			describe appropriate email conventions when communicating in this way.	internet can be used to provide different methods of communication is improving.		
Information technology	- 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 2Quiz example (sorting shapes), 2Code design mode (manipulating backgrounds) or using pictogram software such as 2Count.	- Children demonstrate an ability to organise data using, for example, a database such as 2Investigate and can retrieve specific data for conducting simple searches. Children are able to edit more complex digital data such as music compositions within 2Sequence. 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 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 (2Question), using software such as 2Graph. Children can consider what software is most appropriate for a given task. They can create purposeful content to attach to emails, e.g. 2Respond.	- 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 2Connect and 2Publish+. Children share digital content within their community, i.e. using Virtual Display Boards.	- 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 2Code. 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. 2Blog, Display Boards and 2Email.	- 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. 2Blog. They are able to use criteria to evaluate the quality of digital solutions and are able to identify improvements, making some refinements.
Digital literacy	- 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.	- 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. 2Publish example template. Children make links between technology they see around them, coding and multimedia work they do in school e.g. animations, interactive code and programs. - 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 2Respond activities on Purple Mash and know ways of	- 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 2Email in Purple Mash. They know more than one way to report unacceptable content and contact.	- Children can explore key concepts relating to online safety using concept mapping such as 2Connect. They can help others to understand the importance of online safety. Children know a range of ways of reporting inappropriate content and contact.	- 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 online services. Children implicitly relate appropriate online behaviour to their right to personal privacy and mental wellbeing of themselves and others.	- 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. 2Respond activities. They recognise the value in preserving their privacy when online for their own and other people's safety.

		reporting inappropriate behaviours and content.				