Computing
Curriculum Documents

Intent

At Wedmore First School Academy, we are committed to nurturing 'thinkers of the future' through a modern and engaging computing education. Our aim is to empower children with computational thinking and creativity, enabling them to thrive as active participants in an ever-evolving digital landscape. We believe it is essential for children to harness technology not just as a learning tool but as a means of self-expression and innovation for their generation.

Our comprehensive computing curriculum strikes a balance between acquiring extensive knowledge and practical application across diverse digital contexts. In addition to discrete computing lessons, we provide opportunities for children to integrate their skills into other areas of the curriculum. With a focus on responsible and respectful technology use, we ensure that children are aware of both the benefits and potential pitfalls of online engagement, promoting safe practices and confident use of technology.

Implementation

Our scheme of work for Computing is adapted from the 'ELim' Curriculum and covers all aspects of the National Curriculum. It encompasses three key areas: Computer Science, Information Technology, and Digital Literacy. We employ a spiral curriculum approach, allowing pupils to revisit and build upon key concepts each year.

The curriculum aims to equip young people with the knowledge, skills and understanding they need to thrive in the digital world of today and the future.

The national curriculum for computing aims to ensure all pupils:

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation (Computer science)
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems (Computer science)
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems (Information technology)
- are responsible, competent, confident and creative users of information and communication technology. (Digital literacy)

Impact

In our school, we strive to cultivate confident users of technology among our children. They will be well-equipped to harness digital tools for various goals at home and in school. With a thorough understanding of the implications of technology and digital systems, our children are prepared for a fast-evolving society. Progress in our computing curriculum is evident through positive outcomes, fostering enthusiasm and confidence as they transition to middle school. Our children will emerge as competent and adaptable 'Computational Thinkers', capable of applying recognised concepts across their learning This comprehensive foundation ensures they are prepared for the future in an increasingly digital world.

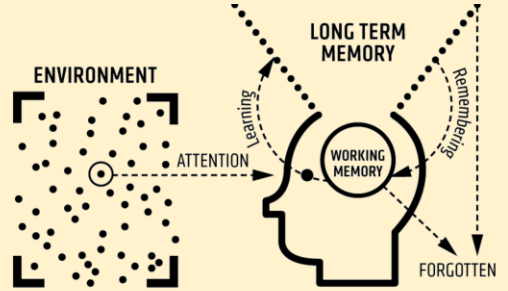
Wessex Learning Trust Principles

Strategic Aims

The Principles codify the shared language that contribute to high-quality, adaptive teaching and inclusion for all.
Used routinely to bring the curriculum to life, the pedagogical principles support learning and progress over time.
The Wessex Principles are not a linear planning tool, an expectation for every lesson or mandate a formulaic approach to lessons

The principles aim to:

- Reduce cognitive load
- Encourage self regulation
- Provide regular opportunities to identify misconceptions or gaps in learning
- Ensure teaching is adapted to need
- Make learning explicit and transferable across the curriculum, beyond school into the wider community and wider world



Ready To Learn Routines → **Linking Prior + New Learning** → **Focused Instruction *'I Do'*** → **Practise Learning *'We Do'*** → **Learning Check *'You Do'*** → **Consolidating Learning**

★ Subject pedagogies are key ingredients to adaptive teaching, alongside effective formative and summative feedback to monitor progress.

★ Disciplinary and substantive learning is integral to any planned sequence of learning.

Ready To Learn Routines		*Ref SLC* - *Emotional learning environment* - *physical learning environment*	Learning environments are safe, inclusive and welcoming. Relationships are positive and love of learning is promoted. Everyone feels safe to take risks and explore learning without judgement. Praise and rewarding effort is used to motivate and engage. A sense of pace and challenge is established from the start of the lesson.
Linking Prior + New Learning		*Ref SLC* - *Pace of talk, clarity of instruction*	Prior learning is checked and revisited to strengthen connections and longer-term memory. Know more, remember more. Planning ensures new learning builds on prior learning. Vocabulary is explicitly taught using the schools agreed pedagogies so that words are understood, contextualized and barriers to learning are reduced. Problem solving and number skills are revisited, retaught and applied in unfamiliar contexts to support deeper learning. Gaps in learning and misconceptions are revisited, including feedback and improvement tasks. Planning is adapted lesson on lesson so that core skills and knowledge are retaught where necessary. Precision learning is explained so that skills and knowledge are well understood, and misconceptions are minimised.
Focused Instruction *'I Do'*		*Ref SLC* - *Explicit teaching of vocabulary* - *Explicit teaching of listening*	The steps to new learning are broken down into manageable amounts and reduce cognitive load. High-quality explanations are used to model thinking, decision making, and application of knowledge. Self-regulation is taught through decision making modelled, visible and explicit. Approaches to getting unstuck are taught and accepted as part of learning. Practical skills and strategies are modelled so that there is a clear understanding of how to solve problems solve and minimize misconceptions. Deeper learning is sequenced so that all learners can understand each developing stage. Learners know what excellent learning looks like and have success criteria to support their independent work.
Practise Learning *'We Do'*		*Ref SLC* - *Explicit teaching paired, small group talk*	Guided practice and worked examples are used to link new learning and decision making with prior learning. Formative assessment, including rich questioning, is used skilfully to check understanding and the impact of planned learning. Peer explanation + modelling scaffolds and prepares for independent practice. Learners use expert thinking and talking to explore deeper learning. Scaffolding and support (including TAs) is in place to develop and build independence.
Learning Check *'You Do'*			Skills and knowledge are explored using a variety of contexts. Independent practice and application of learning (including homework) builds confidence, self esteem and motivation. Metacognition and self-regulation are developed over time. Learning is consolidated. Scaffolding and support is reduced and removed over time. Feedback is used to deepen learning and address misconceptions.
Consolidating Learning			Learner's plan, review and evaluate their progress reflecting on what excellent learning looks like and success criteria. Next steps are identified and used to inform teacher planning and develop mastery approaches over time. Learning skills continue. *Next lessons, rest of day, community, wider world.*

Knowledge in Computing

National Curriculum and EYFS Framework

Substantive Knowledge
Learning about…

- All knowledge from strands of Computer Science, Information Technology and Digital Literacy

Disciplinary Knowledge
Learning how to…

- Terminology, block coding commands, software commands
- Functions, network knowledge, online safety knowledge

Procedural Knowledge
Learning through…

(The methods and processes of computing)

- Formatting, searching, discerning content
- Debugging, data manipulation

The EYFS framework is structured very differently to the national curriculum as it is organised across seven areas of learning rather than subject areas. The aim of this document is to show the skills taught across EYFS and how they feed into national curriculum subjects.

This document demonstrates which statements from the 2020 Development Matters are prerequisite skills for computing within the national curriculum. The table below outlines the most relevant statements taken from the Early Learning Goals in the EYFS statutory framework and the Development Matters age ranges for Three and Four-Year-Olds and Reception to match the programme of study for computing. The most relevant statements for computing are taken from the following areas of learning:

- Personal, Social and Emotional Development
- Physical Development
- Understanding the World
- Expressive Arts and Design

Computing			
Two, three and four year olds	Personal, Social and Emotional Development		• Increasingly follow rules, understanding why they are important.
	Physical Development		• Match their developing physical skills to tasks and activities in the setting.
	Understanding the World		• Explore how things work.
Reception	Personal, Social and Emotional Development		• Show resilience and perseverance in the face of a challenge.
	Physical Development		• Develop their small motor skills so that they can use a range of tools competently, safely and confidently. • Know and talk about the different factors that support their overall health and wellbeing: -sensible amounts of 'screen time'.
	Expressive Arts and Design		• Explore, use and refine a variety of artistic effects to express their ideas and feelings.
ELG	Personal, Social and Emotional Development	Managing Self	• Be confident to try new activities and show independence, resilience and perseverance in the face of challenge. • Explain the reasons for rules, know right from wrong and try to behave accordingly.
	Expressive Arts and Design	Creating with Materials	• Safely use and explore a variety of materials, tools and techniques, experimenting with colour, design, texture, form and function.

	National Curriculum Programmes of Study and EYFS Framework					
	Year 1	**Year 2**	**Year 3**	**Year 4**	**Year 5**	**Year 6**
Computing Science	*1a. Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions. 1b. Create and debug simple programs. 1c. Use logical reasoning to predict the behaviour of simple programs.*		*2a. Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. 2b. Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. 2c. Use logical reasoning to explain simple algorithms and to detect and correct errors in algorithms and programs. 2d. 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. 2e. Appreciate how results are selected and ranked, and be discerning in evaluating digital content.*			
Information Technology	*1d. Use technology purposefully to create, organise, store, manipulate and retrieve digital content.*		*2f. Use search technologies effectively. 2g. 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.*			
Digital Literacy	*1e. Recognise common uses of information technology beyond school. 1f. 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.*		*2h. Understand the opportunities [networks] offer for communication and collaboration. 2i. Be discerning in evaluating digital content. 2j. Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.*			

Learning that...

	Year 1	Year 2	Year 3	Year 4
Algorithms and Programming	Know that algorithms are sequences of instructions. Know how to create a simple unplugged algorithm using everyday language or symbols (e.g. instructions for a Lego model). Know how to create a simple program using algorithms on a digital device, with support (e.g. plan steps to control a Bee Bot). Begin to know how to debug an algorithm (e.g. correct instructions given as an unplugged task or to a BeeBot)	Know that algorithms are sequences of instructions or sets of rules. Know how to create a more complex unplugged algorithm using everyday language or symbols (e.g. how to share sweets). Know where to find the commands to move a sprite. (Scratch Jr) Know how to join blocks together. (Scratch Jr) Know how to run the program they have created. Know how to create an onscreen program using algorithms (e.g. Scratch Jr).	Know that algorithms are sequences of instructions or sets of rules. Know that sequencing commands are step-by step instructions. Know the basic features of Scratch (e.g. sprite, background, blocks) Know the relationship between an event and an action in programming. Know where the main types of blocks are located on Scratch. Know how to create a simple program using a block language, without user interaction (e.g. create a simple animation in Scratch with a sprite, dialogue and background) Know how to use sequences of commands or blocks in on-screen programming, producing an output on the screen (e.g. a simple animation in Scratch).	Know that algorithms are sequences of instructions or sets of rules. Know that repetition commands are repeated instructions that loop until a condition has been met. Know that loops can be count controlled or infinite. Know how to create a program using a block language, with simple user interaction (e.g. create a simple game involving use of backgrounds, props, sprites, costumes, sound). Know how to use sequences & repetition [e.g. repeat… until…] of commands or blocks in onscreen programming, inc keyboard inputs & on-screen outputs (e.g. write a game using Scratch with repeated commands) Know how to use a count controlled loop both in a real life context and on programming software. (e.g. Logo)
Logical Reasoning	Know how to predict what the outcome of giving a command will be. (e.g. to a BeeBot) Know how to predict the outcome of a simple sequence (e.g. a BeeBot sequence) using forwards and backwards). Know how to predict the outcome of a sequence with up to four commands.	Know how to give a logical explanation for predicting the behaviour of programs. (e.g. their Scratch Jr animation)	Know how to explain a sequence algorithm in own words. This could be graphical (e.g. explain the reasoning for a Scratch animation). Know how to use logical reasoning to begin to detect errors in their own or others' programs, giving reasons.	Know that networks are physically connected. Know how information is shared. Know what the World Wide Web is and how it is used. Know that not all information on the internet is reliable. Know that when they are detecting errors in programs, they are debugging. Know how to explain an algorithm using sequence and repetition, in their own words (e.g. explain the algorithm for their Scratch game). Know how to use logical reasoning to detect and fix errors in their own or others' programs, giving reasons, including testing the program to ensure they are fixed.
Networks	Know that there are common uses of information technology beyond school.	Know that there are uses of information technology beyond school, including knowing basic computer systems and networks.	Know what input, output and process mean. Identify input and output devices. Know what a network is. Know that information is shared on a network. Know that devices are connected to a computer. Know what a switch, server and wireless network point are.	

Information Technology – learning how to…					
		Year 1	**Year 2**	**Year 3**	**Year 4**
Using and Creating		• Know the main parts of a computer/laptop. • Know to click and drag using a mouse. • Know the software commands 'open' and 'save' and that they are used to store and retrieve work. • Know how to store and retrieve work. • Recognise keys on a keyboard (including arrow keys, space, backspace and shift) • Begin to recognise some formatting tools (bold, underline) • Begin to know how to select text. Know how to type capital letters using a keyboard. • Recognise shape, line, fill and brush tools on a paint program. • Know how to use the shape, line, fill and brush tools on a paint program and how to change the shape of these. • Know what a label is and how it is used for data collection. • Know how to describe the properties of an object for the purpose of data collection.	• Know how to add text or images on a software package such as Teams or Padlet. • Know how to leave a comment on a software package such as Teams or Padlet. • Know how to place notes, change their pitch and delete notes on Song Maker (Chrome Lab) • Know how to test music on Song Maker (Chrome Lab) • Know how to enter data into a computer for a pictogram. • Know what data is. Know what a label is. Know how to create labels on j2e Pictogram. • Know how to increase/decrease the number of images on j2e Pictogram. Know the software commands 'open' and 'save' and know that naming files appropriately is important for retrieving work. • Know some basic editing commands in a word processing program (e.g. Microsoft Word)	• Know some basic editing commands in a desktop publishing program (e.g. Word, Publisher) • Know the difference between text and images and what they are used for. • Know some page settings in a desktop publishing program (e.g. Word, Publisher) Know how to use basic editing commands in a desktop publishing program. • Know how to change some page settings in a desktop publishing program. • Know what a poll is and how questions can be asked on a poll. • Know how to create a poll in Poll Maker, including how to add images. • Know how to access poll results on Poll Maker. • Know some of the commands in a video editing program. (e.g. Windows Movie Maker, Filmora or iMotion) • Know how to use a device to take a sequence of images or videos for a stop frame animation. • Know how to insert images or videos into video editing software (e.g. Windows Movie Maker, Filmora or iMotion). • Know how to create a stop frame animation using video editing software.	• Know some basic commands in presentation software. (e.g. PPT) • Know how to create a linear presentation using presentation software. (e.g. Powerpoint) • Know the words data, cell, rows and columns and where these can be found on a spreadsheet program. (e.g. Microsoft Excel) • Know some basic formatting commands on a spreadsheet program. (e..g Microsoft Excel) • Know how to format spreadsheets Know how to input basic number operations to work out calculations on a spreadsheets. • Know that data can be presented in different ways on spreadsheets • Know how to insert and format an image on a word processing program. (e.g. Microsoft Word) Know more complex formatting commands on a word processing program. (e.g. Microsoft Word) • Know how to edit & improve the layout of a document on a word processing program. (e.g. Word) • Know how to format and check text on a word processing program. (e.g. Microsoft Word) • Know how to use & combine a variety of software on a computer (e.g. analyse data in spreadsheet and present in Powerpoint). • Know how to design and create content on a computer (e.g. plan, shoot and edit a video, plan and create a presentation) • Know how to collect and present information in different ways. (e.g. collecting data for a branching database)

Information Technology – learning how to...					
		Year 1	Year 2	Year 3	Year 4
Searching			• Know how to search for information more safely using 'for kids'. • Know where to search for images safely. • Know how to click on a weblink.	• Know how to search for information within a single site, using browser-specific tools (e.g. 'find,, 'back') & site-specific tools (e.g. 'search', 'autocomplete'). • Know that search engines select pages according to index of keywords found in the content, and that they rank pages according to relevance	
Tier 2 vocabulary		• Create, organise • Predict, information, personal • Instructions sequence, repetition	• Store, content, private, command, control analyse, research, secure, highlight, selection, error, duration	• Year 3: Precise, application, media, table, template, undo, evident, export, summary, chart, variables, convert, random, schedule • Year 4: input, process, output, design, manipulate, retrieve, percent, relevant, amend, attach, ambiguous, bias, terminate, compile, function	
Tier 3 Vocabulary		• Technology internet, online, blog, icon, login • Document, data, file, folder • Code, bug	• Device, app, computer, download, database, font, presentation, debug, loop	• Year 3: buffer, cookie, email, upload, backup, configure, filter, spreadsheet, algorithm, programme, flowchart • Year 4: digital, analogue, graphics, compress, dashboard, hyperlink, virus, simulation, decompress, conditionals	

Digital Literacy – Substantive and Disciplinary Knowledge

	Year 1	Year 2	Year 3	Year 4
Self-image and Identity	• Know that there may be people online who could make them feel sad, embarrassed or upset. Give examples of when and how to speak to a trusted adult.	• Explain and describe how other people's identity online can be different to their identity in real life. • Give examples of issues online that might make them feel sad, worried, uncomfortable or frightened; give examples of how they might get help.	• Understand 'identity' and explain they can represent themselves online in different ways; explain ways in which and why they might change their identity depending on what they are doing online (e.g. gaming; using an avatar; social media).	• Explain how their online identify can be different to the identity they present in 'real life'; knowing this, describe the right decisions about how to interact with others and how others perceive them.
Online Relationships	• Use the internet with adult support to communicate with people they know. • Explain why it is important to be considerate and kind to people online	• Use and exemplify ways that the internet can be used to communicate with people they don't know well (e.g. email a penpal in another school/ country).	• Explain why they should be careful who they trust online and what information they give, and explain some risks communicating online with others they don't know well. • Explain what it means to 'know someone' online and why this might be different from knowing someone in real life. • Explain how to be a good digital citizen.	• Describe strategies for safe and fun experiences in a range of online social environments. • Give examples of how to be respectful to others online.
Online Reputation	• Describe what information they should not put online without asking a trusted adult first.	• Explain how information put online can last a long time. • Know who to talk to if they think someone has made a mistake about putting something online.	• Know how to search for information about themselves online. • Recognise they need to be careful before putting info about themselves of others online. • Know who to ask if they are not sure if they should put something online.	• Explain how to keep online games fun and friendly.
Online Bullying	• Begin to know how to behave online, in ways that do not upset others. • Explain what to do if they feel sad or worried about something online.	• Give examples of bullying behaviour online, understand how it can make people feel and talk about how someone could get help online or offline.	• Explain what bullying is and can describe how people may bully others. • Know how to behave online.	• Identify some online technologies where bullying might take place and describe ways people can be bullied through a range of media (e.g. image, video, text, chat). • Explain why they need to think carefully about how content they post might affect others, their feelings and how it may affect how others feel about them (their reputation)

Digital Literacy –Substantive and Disciplinary Knowledge

	Year 1	Year 2	Year 3	Year 4
Health, wellbeing and lifestyle	• Describe and explain rules to keep them safe when using technology both in and beyond the home.	• Describe and explain rules to keep them safe when using technology both in and beyond the home, and say how these rules guide them	• Describe and explain rules to keep them safe when using technology both in and beyond the home, and say what advice they could give to others to stay safe online.	• Explain how using technology can distract from other things that should or need to be done. • Identify times and situations when technology use may need to be limited, and suggest strategies for doing this.
Privacy and security	• Recognise examples of personal information (e.g. name, date of birth, family's names, school). • Explain why they should always ask a trusted adult before sharing any personal information online.	• Describe how online personal information could be seen by others. • Describe and explain some rules for keeping information private. • Explain what passwords are and use passwords for accounts and devices.	• Explain how to create a strong password and how to keep this safe.	• Explain that others online can pretend to be them or other people, including friends, and suggest reasons why they might do this. • Explain how we keep our personal information safe.
Copyright and ownership	• Explain why work they create belongs to them and save it so that others know it belongs to them.	• Describe why other's work belongs to them, and recognise that content on the internet may belong to other people.	• Explain why copying someone else's work from the internet without permission can cause problems, and give examples of these problems.	• Explain why they need to consider who owns content that is searched for, whether they have the right to use it, and give examples.
Communication and Collaboration	• Know how to work collaboratively to create a set of instructions linked to algorithms.	• Know how to work collaboratively on a class blog page, such as using a Teams channel or Padlet.		• Know how to work collaboratively on a class blog page, such as using a Teams channel or Padlet.

Knowledge Glossary

Subject Knowledge and Terminology Required for Understanding National Curriculum Statements

Key Stage 1 Computer Science [CS]

Understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions.: An algorithm is a precisely defined procedure – a sequence of instructions for performing a specific task. Computer programs, like algorithms, are comprised of sets of rules or instructions, but they differ in that they need to be written in a precise language a computer can 'understand'. A computer's central processor understands a very limited set of simple instructions written in machine code. Very few programmers work at this level, so computer scientists have developed programming languages, which sit somewhere between the ideas in the algorithm and the computer's machine code. There are many different programming languages, each having their own vocabulary, grammar and features that make them good for particular tasks. The current favourites in primary schools are Scratch, Logo and Kodu.

Create and debug simple programs: The best way for pupils to learn what an algorithm is, and how it can be implemented as a program, is to write some programs. Programming involves taking an idea for doing something and turning it into instructions the computer can understand. In the infant classroom this could be writing a set of commands for a Bee-Bot, ProBot or Roamer, or snapping on-screen program building blocks together in Scratch. When you write a program you need to have a clear idea of what it will do and how it should do it. This is where algorithms come in, and thinking algorithmically is an integral part of the craft of programming. Most programs don't work as they should first time round; professional programmers have this experience all the time! One of the most rewarding aspects of programming is finding and fixing these mistakes. Mistakes in programs are called 'bugs', and finding and fixing them is 'debugging'. The process of debugging often involves identifying that there is a fault, working out which bit of the program (or underlying algorithm) has caused the problem, and then thinking logically about how to fix it. In the classroom, this can provide a great opportunity for collaborative work. As a teacher, you should identify clear steps that pupils can follow so that they can fix their code. These might involve identifying what the fault is, finding out which part of the code is creating the problem, and then working towards a fix.

Use logical reasoning to predict the behaviour of simple programs: Computers are deterministic machines. We can predict exactly how they'll behave through repeated experience or by developing an internal model of how a piece of software works. Stepping through the program can give a clear sense of what it does, and how it does it, giving a feel for the algorithm that's been implemented. In the classroom, getting one pupil to role-play a floor turtle or screen sprite while another steps through the program can give a far more immediate sense of what's going on. When working with a computer, encourage pupils to make a prediction about what the program will do before they press return or click the button, and to explain their prediction logically; this is part of computer science. Logical reasoning also implies that pupils are following a set of rules when making predictions.

Key Stage 1 Information Technology [IT]

Use technology purposefully to create, organise, store, manipulate and retrieve digital content: Creating digital content has many practical possibilities. These include commonplace tasks such as word-processing, creating pictures using paint packages, working with digital photographs and video, writing computer programs, and creating online content such as blog posts, forum contributions, wiki entries and social network updates. This creative work is digitised (i.e. converted to numbers) once it's on the computer. The sheer quantity of digital information makes the skill of organising digital content more important than ever. In more practical terms, we might think of how to bring together different digital media, how to order a series of paragraphs, how to organise the files in our documents directory, or how to tag photos and posts online. Knowing where a file is saved in the directory structure is important. It's vital to be able to distinguish between the hard disk (or solid state storage) inside the computer itself, the school's network server, USB disks or memory cards, and online storage via the internet. Manipulating digital content is likely to involve using one or more application programs, such as word-processors, presentation software, or image-, audio- or video-editing packages. The pupil makes changes to the digital content. The skill here is not just using the software tools, but also knowing how best to change the content for the audience and purpose, and to take into account principles of good design. Retrieving digital content could be seen as the reverse of storing: the skills of opening and saving documents are similar. Retrieving content requires you to know what you called the file, what file type it is, and where you stored it. Finding files can be time-consuming, especially when the filing system is not well organised. Computer filing systems have search features to make this easier, but are reliant on the user remembering enough about the file to be able to search.

Key Stage 1 Digital Literacy [DL]

Recognise common uses of information technology beyond school: Digital technology is a part of all our lives. A key stage 1 pupil might be woken by a digital alarm clock, have a bowl of microwaved porridge for breakfast, and then watch digital TV or play an iPad game. While they're at school, their attendance, progress and lunch are tracked through the management information system, they engage in activities on tablets, and research things on the web. The ingredients for the evening meal may have been ordered online, or a parent may have scanned them at the supermarket. There are many opportunities for pupils to consider the applications of algorithms, programs and systems.

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: This statement covers the key principles of pupils' online safety. Pupils should be aware of the main risks associated with the internet, and recognise that they should not share certain types of personal information online. Pupils should have an age-appropriate understanding of their responsibilities under the school's acceptable use policy.

Knowledge Glossary

Subject Knowledge and Terminology Required for Understanding National Curriculum Statements

Key Stage 2 Computer Science [CS]

Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts: The focus on algorithms at key stage 1 leads pupils into the design stage of programming at key stage 2. Algorithms are the necessary start of the process of creating working code, and identifying the steps needed to solve any problem is essential. Splitting problems into smaller parts is part of computational thinking. For example, designing a game in Scratch will involve thinking about algorithms, programming, drawing sprites and backgrounds, making animations, and even composing music or recording sound effects.

Computing Concepts - Use sequence, selection, and repetition in programs; work with variables and various forms of input and output:

- Sequence in this context is the step-by-step nature of computer programs, mirroring the sequence of steps the algorithm would list.
- Selection refers to instructions such as if ... then ... otherwise decisions in which the operation (what the program does) depends on whether or not certain conditions are met. For example, a quiz provides different feedback if the player answers the question correctly or incorrectly. It is helpful to refer pupils to selections (choices) they make in everyday life; for example, if it rains in the morning, then I will wear my anorak to school, otherwise I won't.
- Repetition is a programming structure such as a repeat ... until loop in which the computer runs part of the program a certain number of times or until a particular condition is met. In the case of the quiz, we might want to ask ten questions, or keep going until the player has scored five correct answers. Again, it is useful to refer pupils to loops or repetition in daily routines. For example, the traffic lights on a pelican crossing will stay green until someone presses the button to cross the road.
- Variables are used to keep track of the things that can change while a program is running. They are a bit like x or y in algebra, in that the values may not initially be known. Variables are not just used for numbers. They can also hold text, including whole sentences ('strings'), or the logical values 'true' or 'false'. For our quiz we would use variables to keep track of the player's score and the number of questions they attempt. Variables are like boxes, in that the computer can use them to store information that can be changed by the user, the program or by another variable.

We may think of input as keyboard and mouse (or touch screen), and output as the computer display, but pupils' experiences should be widened beyond this. Working with sound is straightforward, as laptops have built-in microphones and speakers. The latest version of Scratch provides support for using webcams. Digital cameras allow interesting work using image files. The reference in the programme of study to 'controlling physical systems' implies the use of sensors, motors and perhaps robotics. Midi instruments like an electronic keyboard, and devices such as Lego WeDo, MaKey MaKey and Microsoft Kinect provide yet further experience of working with various forms of input.

Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs: Key stage 2 pupils should be able to explain the thinking behind their algorithms, talking through the steps and explaining why they've solved a problem the way they have. They also need to be able to look at a simple programming project and explain what's going on. This is made easier with languages like Scratch, Kodu and Logo, which feature an on-screen sprite or turtle. The immediate feedback helps pupils to understand and debug their programs. Pupils might also be expected to look at someone else's algorithm and explain how it does what it does.

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: This is a challenge because most of us have not thought about how these ever-present technologies do what they do. Computer networks, including the internet, are made up of computers connected together. The computers include fast, dedicated machines that pass on data that's not intended for them (called 'routers', 'gateways', 'hubs' or 'switches', depending on particular roles), and 'servers' (always-on machines looking after emails, web pages and files that other computers might ask for from time to time). The connections between the computers in a network may consist of radio or satellite signals, copper wires or fibre-optic cables. Information stored on computers and information travelling over networks must be digitised (i.e. represented as numerical data). The computer network in your school and the internet use the same method or 'protocol' to send and receive this data. The data is broken up into small 'packets', each with identifying information, which includes the IP (internet protocol) address of the sender and recipient. These packets of information make their way across the internet from source to recipient. At the far end, the packets get stitched back together in the right order and the email is delivered, the website is accessed, or the Skype call gets connected. Many of these packets, travelling at near light-speed, are generated by web servers returning web pages to the browser requesting them. By connecting people around the world and passing on packets of data from sender to recipient, the internet has created many opportunities. These range from communication (such as email, video conferencing, blogs, forums, social networks) and collaboration, such as wikis (including Wikipedia), to real-time collaborative editing, Creative Commons media (permission to share and use creative work with conditions stated by the creator) and open-source software, which is available for us to use and change.

Knowledge Glossary

Subject Knowledge and Terminology Required for Understanding National Curriculum Statements

Key Stage 2 Information Technology

Use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content: Using search technologies involves aspects of computer science, information technology and digital literacy. Effective use of search engines gets the results you want. It relies on specifying the right keyword, skimming and scanning the results to see which seems most relevant, and distinguishing between the main results and adverts presented as sponsored results. It may also involve using other features7 of the search engine, including searching for phrases rather than keywords, or limiting searches to a particular time frame, language, reading level or website. Search engines take many factors into account. At the heart of Google's algorithms8 is 'PageRank', which determines the quality and rank of a page based on the quality of the pages that link to it. Their quality is, in turn, determined by the quality of the pages that link to them, and so on. Just because a page has a high rank in Google or another search engine for a particular query, it doesn't mean that the content is true, age appropriate or relevant to a particular project. Pupils need to develop skills in evaluating digital content, including how trustworthy the information is (perhaps by verifying it with another independent source), whether it's something that the audience for a project would be able to grasp, and why the content was posted in the first place (e.g. to give a balanced overview, or simply to advance one side of an argument).

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: This is something of a catch-all requirement, bringing together various aspects of the computing curriculum. Pupils might typically be expected to demonstrate progression by: using software under the control of the teacher; moving onto using software with increasing independence; then, combining software (e.g. importing an edited image or video into a presentation or web page); and then, selecting software themselves (perhaps from the full range of applications installed on computers, smartphones and tablets at home or at school, or available to them via the web). Internet services might include, for example, learning platforms, school, class or individual blogs, and cloud-based tools such as Google Drive, Office 365 or image-editing sites. The reference to 'a range of digital devices' encompasses using both fixed and mobile technologies. It also includes running software (such as that described in the previous paragraph) on web servers via the internet. There is also recognition that design and creativity in computing encompass many forms, from the content familiar to many from the old ICT programme of study, the programming as required by earlier statements in the new programme of study, to more complex, system-level ideas, combining software and hardware to achieve a well-defined goal with a particular audience in mind. At key stage 2 it might be more helpful to think of data as numbers and information as richer media such as text, images, audio, and video or 3D representations. However, it is worth remembering that both data and information are digitised by computers (i.e. stored in the form of numbers). Collecting, analysing, evaluating and presenting data is an important application of computers. Pupils should gain experience of working with data they have generated or collected for themselves, as well as big, public datasets.

Key Stage 2 Digital Literacy [DL]

Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact: Safe and responsible use of technology at key stage 2 builds on skills learned in key stage 1. As well as requiring pupils to keep themselves safe and to treat others with respect, the programme of study at key stage 2 introduces an emphasis on responsible use of technology. Pupils need to consider how their online actions impact other people. They need to be aware of their legal and ethical responsibilities, such as showing respect for intellectual property rights (e.g. musical, literary and artistic works), keeping passwords and personal data secure, and observing the terms and conditions for web services they use (such as the 13+ age restriction on Facebook). Pupils should also develop some awareness of their digital footprint: the data automatically generated when they use the internet and other communication services, and how this is, or could be, used. Pupils should be aware of, and abide by, the school's acceptable use policy, as well as the requirements of any other services they use. Encourage pupils to think twice, and to check terms and conditions, before signing up for internet-based services.