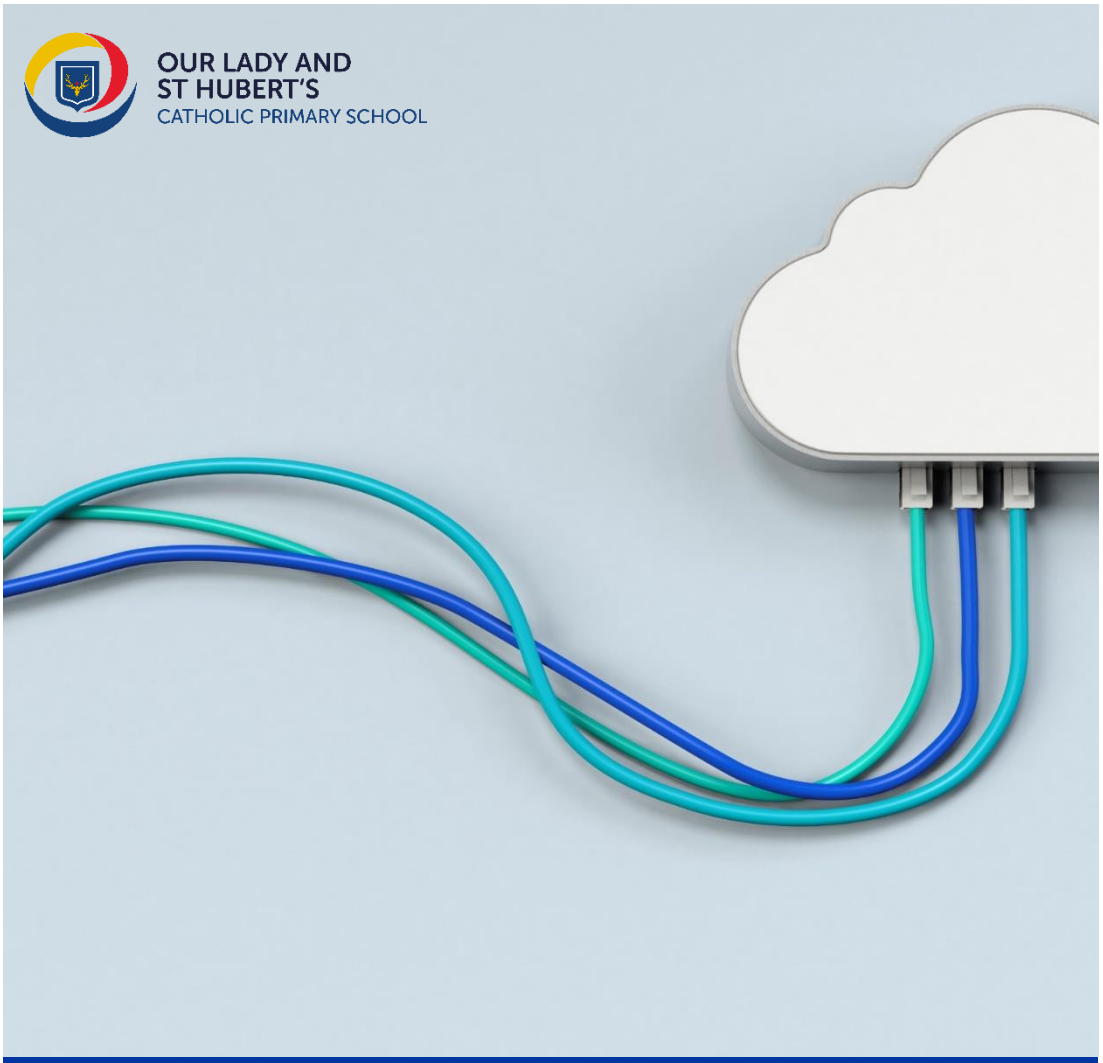
OUR LADY AND
ST HUBERT'S
CATHOLIC PRIMARY SCHOOL



At Our Lady and St. Hubert's, home, school and parish work together, knowing that God is with us in all we do.

Computing

Guidance and Procedures

Contents

Computing Curriculum Intent, Implementation and Impact	Page 2
Pedagogy	Page 4
Digital Literacy	Page 5
Role of the Subject Leader	Page 6
Curriculum Planning and Expectations	Page 6
Computing and the 6Cs	Page 6
Feedback and Assessment	Page 7
SEND, Pupil Premium and Inclusion	Page 8
Adapting the Curriculum	Page 8

Intent

At Our Lady and St Hubert's we want pupils to be *masters* of technology and not slaves to it. Technology is everywhere and will play a pivotal part in students' lives. Therefore, we want to model and educate our pupils on how to use technology positively, responsibly, and safely. We want our pupils to be creators not consumers and our broad curriculum encompassing computer science, information technology and digital literacy reflects this. We want our pupils to understand that there is always a choice with using technology and as a school we utilise technology to model positive use. We recognise that the best prevention for many issues we currently see with technology/social media is through education. Building our knowledge in this subject will allow pupils to effectively demonstrate their learning through creative use of technology. We recognise that technology can allow pupils to share their learning in creative ways. We also understand the accessibility opportunities technology can provide for our pupils. Our knowledge rich curriculum is balanced with the opportunity for pupils to apply their knowledge creatively, which will in turn, help our pupils become skilful computer scientists. We encourage staff to embed computing across the whole curriculum to make learning creative and accessible. We want our pupils to be fluent with a range of tools to best express their understanding and by Upper Key Stage 2, children will have the independence and confidence to choose the best tool to fulfil the task and challenge set by teachers.

Implementation

Our knowledge progression for computing is ambitious. We recognise that to achieve our intent for computing, this intent must be implemented using current academic research – research in both computing and cognition. Within the *Computing Curriculum*, every year group learns through units within the same four themes (Computer systems and networks, programming, data and information and creating media), which combine the ten strands of the *National Centre for Computing Education's* taxonomy (See Below).

Primary themes	Computing systems and networks	Programming	Data and information	Creating media
Taxonomy strands	Computer systems Computer networks	Programming Algorithms Design and development	Data and information	Creating media Design and development
	Effective use of tools			
	Impact of technology			
	Safety and security			

To develop a rich and varied schema in our children's brains, our progression has been developed so that learning is sequential, allowing knowledge and skills to be built upon. Key learning objectives are delivered to pupils in small steps, avoiding *cognitive overload* – allowing knowledge to enter long-term memory more readily – therefore allowing **all** children to know more and remember more. We also recognise that over time, this knowledge can be lost – best shown by *'The Ebbinghaus Curve'* and therefore it is important to continually review and retrieve this knowledge. Our curriculum is structured to allow for *spaced learning* and continual retrieval of taught information – creating a *spiral curriculum.* This means that each of

the themes is revisited regularly (at least once in each year group), and pupils revisit each theme through a new unit that consolidates and builds on prior learning within that theme. This style of curriculum design reduces the amount of knowledge lost through forgetting, as topics are revisited yearly. It also ensures that connections are made even if different teachers are teaching the units within a theme in consecutive years.

Children will learn fundamental knowledge of various aspects of computing, revisiting and building their knowledge of key ideas and developing mastery by applying their learning in practical problem-solving contexts.
They will learn how different hardware and software can help them to learn and express their ideas, applying this growing understanding into their wider curriculum learning.
They will collaborate and share their learning using online tools, developing an awareness of how to behave respectfully and positively and keep themselves safe.
They will design, create and test programs and physical systems to solve real-world problems and express their creative ideas.

Knowledge Organisation (Taxonomy Strands)

The *Computing Curriculum* uses the National Centre for Computing Education's computing taxonomy to ensure comprehensive coverage of the subject. All learning outcomes can be described through a high-level taxonomy of ten strands, ordered alphabetically as follows:

Algorithms (AL)—Be able to comprehend, design, create and evaluate algorithms
Computer networks (NW) —Understand how networks can be used to retrieve and share information, and how they come with associated risks
Computer systems (CS)— Understand what a computer is, and how its constituent parts function together as a whole
Creating media (CM)— Select and create a range of media including text, images, sounds and video
Data and information (DI) —Understand how data is stored, organised, and used to represent real-world artefacts and scenarios
Design and development (DD) —Understand the activities involved in planning, creating, and evaluating computing artefacts
Effective use of tools (ET) —Use software tools to support computing work
Impact of technology (IT) —Understand how individuals, systems, and society as a whole interact with computer systems
Programming (PG) —Create software to allow computers to solve problems
Safety and security (SS) —Understand risks when using technology, and how to protect individuals and systems

The taxonomy provides categories and an organised view of content to encapsulate the discipline of computing. Whilst all strands are present at all phases, they are not always taught explicitly.

For the reasons identified above, we teach computing on a regular basis. We know that children engage more- and retain more- when they can make connections and links between their learning, so teachers will ensure that, where appropriate, computing lessons will be linked to half-termly themes and computing skills are applied where appropriate – *interleaving* concepts to develop schema. *Interleaving* is a **method of teaching where students learn concepts in different ways at different times**. For example, children may create presentations (including aspects such as voice overs and animations) to demonstrate their understanding in geography, history or science.

Underpinning our lessons will be our 6Cs - *'21st Century skills for effective learning'*, which will help to shape the lessons planned by our teachers, building on skills such as; communication, resilience, collaboration, critical thinking, creative problem solving and living as an active global citizen – all skills that can be demonstrated through our computing progression.

Impact

Students will become confident users of technology, understanding how digital tools can empower them to work more effectively. They will be able to select and combine applications to help them realise their creative visions. They will become creators and not consumers.

Students will be able to solve real-world problems, thinking about problems logically and designing, realising and testing solutions to these.

Students will be able to navigate confidently online, knowing how to find and scrutinise information, share and collaborate, and protect themselves and others. Finding the right balance with technology is key to an effective education and a healthy lifestyle. We feel the way we implement computing helps children realise the need for the right balance and one they can continue to build on in their next stage of education and beyond.

Pedagogy

Our pedagogical approach consists of 12 key principles underpinned by research: each principle has been shown to contribute to effective teaching and learning in computing and examples of their application can be found throughout the units of work at every key stage.

- ✓ Lead with concepts
  Support pupils in the acquisition of knowledge, through the use of key concepts, terms. and vocabulary, providing opportunities to build a shared and consistent understanding. Glossaries, concept maps and displays, along with regular recall and revision, can support this approach.
- ✓ Work together
  Programming and peer instruction, and also structured group tasks. Working together stimulates classroom dialogue, articulation of concepts, and development of shared understanding.
- ✓ Get hands-on
  Use physical computing and making activities that offer tactile and sensory experiences to enhance learning. Combining electronics and programming with arts and crafts (especially through exploratory projects) provides pupils with a creative, engaging context to explore and apply computing concepts.
- ✓ Unplug, unpack, repack
  Teach new concepts by first unpacking complex terms and ideas, exploring these ideas in unplugged and familiar contexts, then repacking this new understanding into the original concept. This approach. called 'semantic waves', can help pupils develop a secure understanding of complex concepts.
- ✓ Model everything
  Model processes or practices —everything from debugging code to binary number conversions — using techniques such as worked examples and live coding. Modelling is particularly beneficial to novices, providing scaffolding that can be gradually taken away.
- ✓ Foster program comprehension

Use a variety of activities to consolidate knowledge and understanding of the function and structure of programs, including debugging, tracing, and Parson's problems. Regular comprehension activities will help secure understanding and build connections with new knowledge.

- ✓ Create projects
  Use project-based learning activities to provide pupils with the opportunity to apply and consolidate their knowledge and understanding. Design is an important, often overlooked aspect of computing. Pupils can consider how to develop an artefact for a particular user or function, and evaluate it against a set of criteria.
- ✓ Add variety
  Provide activities with different levels of direction, scaffolding, and support that promote learning, ranging from highly structured to move exploratory tasks. Adapting your instructions to suit different objectives will help keep all pupils engaged and encourage greater independence.
- ✓ Challenge misconceptions
  Use formative questioning to uncover misconceptions and adapt teaching to address them as they occur. Awareness of common misconceptions alongside discussion, concept mapping, peer instruction, or simple quizzes can help identify areas of confusion.
- ✓ Make concrete
  Bring abstract concepts to life with real-world, contextual examples, and a focus on interdependencies with other curriculum subjects. This can be achieved through the use of unplugged activities, proposing analogies.
- ✓ Structure lessons
  Use supportive frameworks when planning lessons, such as PRIMM (Predict, Run, Investigate, Modify, Make) and Use-Modify-Create. These frameworks are based on research and ensure that differentiation can be built in at various stages of the lesson.
- ✓ Read and explore code first
  When teaching programming. focus first on code 'reading' activities, before code writing. With both block-based and text-based programming, encourage pupils to review and interpret blocks of code Research has shown that being able to read. trace, and explain code augments pupils' ability to write code.

Digital Literacy

Today's children and young people are growing up in a digital world. As they grow older, it is crucial that they learn to balance the benefits offered by technology with a critical awareness of their own and other's online behaviour and develop effective strategies for staying safe and making a positive contribution online.

It is fundamentally important that children are educated to understand that the digital world in which they live, can open so many avenues in terms of their futures – it is so powerful. However, there are many risks to safety and security online. It is often not the internet and the things that we access that are dangerous – it is how we use it.

At Our Lady and St Huberts, *digital literacy* is more than just one day in February (safer internet day), it is part of our children's understanding beyond this day - it is a part of their everyday lives. We also recognise that for our children to be safe and make a positive contribution online, we must go deeper and further than what is outlined in the *National Curriculum for Computing.*

Our *Digital Literacy* knowledge progression is broken down into the following areas: Self-image and identity; online relationships; online reputation; online bullying; managing online information; health, wellbeing and

lifestyle; privacy and security; and copyright and ownership – in line with *'Education for a Connected World'*. Content is delivered on a regular basis not only computing lessons, but across the curriculum, for our children to access content that empowers them for their future, but in a way that lessens any potential negative impact.

Our framework describes the skills and understanding that children and young people should have the opportunity to develop, at different ages and stages. It highlights what a child should know in terms of current online technology, its influence on behaviour and development, and what skills they need to be able to navigate it safely.

Our progression is taken from the *Education for a Connected World Document.* Using the *Project Evolve* toolkit and knowledge map, teachers plan effectively, basing subsequent lessons on areas of development that have been identified prior to lessons being planned.

Role of the Subject Leader

The Computing Leader will provide vision and guidance for the school on the teaching of Computing. They will be responsible for curriculum design and overseeing delivery. They will provide guidance regarding planning, teaching and assessing the subject and organise CPD where necessary. They will monitor curriculum coverage and teaching of the subject; assessment and analyse attainment data in order to continually review and improve the curriculum and its delivery. They will develop wider opportunities and promote the subject within the school community. They will liaise with all stakeholders in order to ensure Computing at Our Lady and Saint Hubert's fulfils the intent set out in this guidance document.

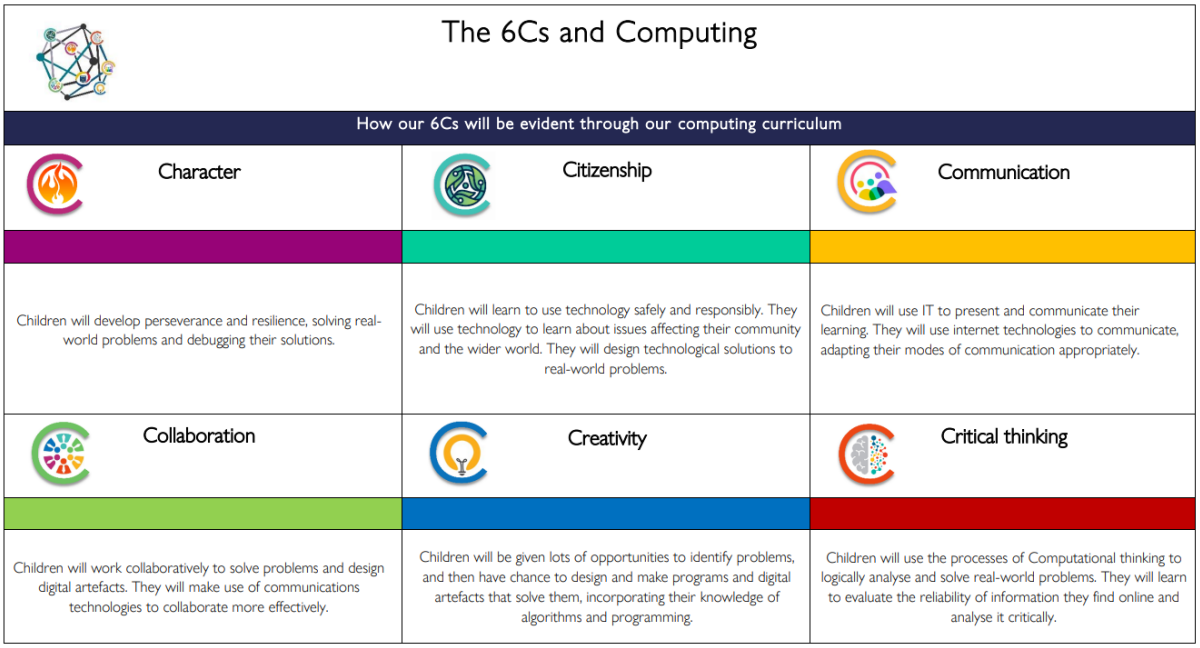
Curriculum Planning and Expectations

The long-term curriculum overview should be used to determine when to teach units. The curriculum overview provides all objectives that need to be covered by each year group. The *'Computing Knowledge Progression'* document provides guidance on how knowledge is broken down into themes of Creating media, Programming, Computer Systems and Networks and Data and Information. This knowledge is progressive and is built upon during children's time in school. Short term planning for each lesson is created from the computing knowledge progression document.

Computing and the 6Cs

To succeed in the 21st century we understand that the children need to be taught, and have time to develop, a different set of skills that is often not catered for in the National Curriculum. We have adapted our curriculum to centre around the 6Cs of 21st Century learning, which are creativity, character, citizenship, critical thinking, collaboration and communication. We provide opportunities for children to persevere and have tenacity to not give up; supporting children to welcome failures, not as set-backs, but opportunities to learn; we encourage children to think critically and challenge the world around them- and then play an active role within it. By teaching these skills and providing opportunities to explore them, we believe that we are providing children with a more rounded school experience, focused on their lives and the world they live in. These skills are as important in computing and the following are examples of where the 6Cs can be applied in Computing:

Character: Children will develop perseverance and resilience, solving real-world problems.
Critical thinking: Children will use the processes of Computational thinking to logically analyse and solve real-world problems. They will learn to evaluate the reliability of information they find online and analyse it critically.
Creativity: Children will be given lots of opportunities to identify problems, and then have chance to design and make programs and digital artefacts that solve them, incorporating various their knowledge of algorithms and programming.
Communication: Children will use IT to present and communicate their learning. They will use internet technologies to communicate, adapting their modes of communication appropriately.
Collaboration: Children will work collaboratively to solve problems and design digital artefacts. They will make use of communications technologies to collaborate more effectively.
Citizenship: Children will learn to use technology safely and responsibly. They will use technology to learn about issues affecting their community and the wider world. They will design technological solutions to real-world problems.



The 6Cs and Computing		
How our 6Cs will be evident through our computing curriculum		
Character	Citizenship	Communication
Children will develop perseverance and resilience, solving real-world problems and debugging their solutions.	Children will learn to use technology safely and responsibly. They will use technology to learn about issues affecting their community and the wider world. They will design technological solutions to real-world problems.	Children will use IT to present and communicate their learning. They will use internet technologies to communicate, adapting their modes of communication appropriately.
Collaboration	Creativity	Critical thinking
Children will work collaboratively to solve problems and design digital artefacts. They will make use of communications technologies to collaborate more effectively.	Children will be given lots of opportunities to identify problems, and then have chance to design and make programs and digital artefacts that solve them, incorporating their knowledge of algorithms and programming.	Children will use the processes of Computational thinking to logically analyse and solve real-world problems. They will learn to evaluate the reliability of information they find online and analyse it critically.

Feedback and Assessment

In line with the school feedback policy, it is not expected that staff will 'mark' work. However, we recognize that feedback is a vital part in children's learning and progression. Staff should therefore look to provide feedback verbally in an appropriate time period. Where learning has been saved to digital exercise books, staff should use the most appropriate form of feedback – this could be a written comment, but voice notes may also be appropriate.

Assessment is the responsibility of teachers. Ongoing summative assessment should be recorded on *Arbor* using the curriculum objectives outlined. Teachers should keep evidence of learning, in the most appropriate way. It is the responsibility of the subject leader to monitor this assessment.

SEND, Pupil Premium and Inclusion for all

It is our belief that **all** children, should have the opportunity to develop across the curriculum including in Computing. Indeed, it is our belief that IT can actively provide accessibility to some areas of the curriculum for children in our care.

Teachers should, therefore:

- Ensure that all children can access learning for Computing.
- Overcome barriers to learning through the use of adaptation and support.
- Provide suitable challenges for more able children, as well as support for those who have emerging needs.
- Respond to the diversity of children's social and cultural backgrounds.
- Actively look for where technology can be used to support children in their learning journey (such as use of the dictation function to support those with difficulties in writing).

Adapting the Computing Curriculum

The Computing Curriculum has been written to support all pupils, with units containing a number of scaffolding activities and utilising effective pedagogies to ensure high quality teaching. However, adaptations may need to be made to enable some pupils, for example those with special educational needs and disabilities (SEND), to access lessons fully. The following principles can be used to help make adaptations that benefit all learners. It is important therefore to identify the needs pupils need support with. For example – A child has poor working memory that means that following instructions is more difficult.

1. Identify essential learning and misconceptions: Determine the key learning in each unit that every child should know. Provide repeated opportunities for pupils to revisit this content in different ways. Identify any likely misconceptions and address these explicitly in lessons. For example, in the year 1 Moving a Robot unit, pupils might struggle with right and left turns and what this looks like for the Bee-Bot, so it is worth spending extra time modelling and practising this.

2. Pre-teach key vocabulary: Pre-teach the essential vocabulary for each unit, provide learners with a word list supported by images and use the vocabulary regularly throughout the unit with a consistent definition. Concentrate on a small number of terms and consider using a graphic organiser to highlight relationships between concepts, e.g. the Frayer model.

3. Create step-by-step instructions: Break down complex tasks and routine skills for using software and hardware into smaller steps and create pictorial instructions for children to follow. For example, in the year 2 Digital Music unit, you can adapt the Chrome Music Lab song maker help card handout to create a sequence of instructions for making their own composition.

4. Provide templates: In Creating Media or Data & Information unts, support task completion by providing a template for pupils to modify – removing the fear of the blank page and helping to build confidence. For example, in the year 2 Pictograms unit, pupils can continue to use the minibeast template rather than set up their own pictogram.

5. Consider non-computing barriers: Consider if difficulties in other areas, such as writing or maths, present barriers to completing a task and if so, modify the task to help mitigate these. For example, in the 'What can you tell me' task within the year 2 Pictograms unit, allow pupils to dictate sentences into a digital version of the worksheet rather than writing them down.

6. Use the PRIMM framework or Parson's problems: In programming units, add extra scaffolding using PRIMM and Parson's problems (A Parson's Problem is a task in which learners are given all of the blocks or lines of code needed to solve a problem, however, the lines have been jumbled so that they are no longer in the correct order. Learners are asked to reorganise the code into the correct order to perform a specific task.). Some pupils may not be able to create a program, but they can practise reading and exploring code in a working program, then modify it to make it more personalised. For example, in the year 1 Programming Animations unit you could provide learners with the code to make the rockets move as a Parson's problem to put together in the correct order. The aim is to remove these scaffolds as children develop their skills, but some learners may not become fully independent.

7. Harness pupils' special interests: Increase engagement and make learning more relevant by incorporating pupils' special interests. This is also important in terms of culturally relevant pedagogy. For example, in the year 1 Programming Animations unit, pupils could make different animals race, or another form of transport.

8. Use unplugged activities and the semantic wave: We can use unplugged activities to help make computing concepts more relevant and understandable for learners. However, it is very important to 'repack' the knowledge of the abstract concept so that learners understand what it means in a wider context and they can use the technical language. For example, in the year 1 Moving a Robot unit pupils work in pairs to develop their understanding of algorithms. One pupil directs their 'robot' partner by giving precise instructions on where to move. Children then need to see the link between instructing their 'robot' partner and how this transfers to pressing buttons on a Bee-Bot, whilst also practising using the key language in context. Pupils with SEND may need repeated examples and smaller steps to repack the knowledge.

9. Support planning: Break down the planning process into smaller parts which can be ticked off as each one is completed, and provide a planning scaffold for learners where required. For example, when planning out algorithms for the Bee-Bot, use a scaffold with the clear and go commands already included and provide command cards to slot into the gaps in between.

10. Reinforce digital skills: A significant barrier to accessing the whole computing curriculum is a lack of key digital skills, for example being able to log on to a computer and use the keyboard effectively. Time spent revisiting digital skills across all units is important to develop fluency. Some pupils may also benefit from extra time to practise these skills in small groups, or may need image-supported help sheets to support specific repeated tasks, such as saving work.

It is important that your adaptations are informed by effective formative assessment to identify any gaps in learning and the approach which may support with these. A further resource which can support you is the Universal Design for Learning Framework from CAST which outlines a number of approaches to include all learners in lessons by providing multiple means of engagement, representation, action and expression.

Reviewed – June 2024 – Anthony Brown

Date of next Review – June 2025