

Chiltern Primary School Computing Progression Map

Together, over time, we are proud to learn to make a difference, to ourselves and others.

Our Curriculum Vision:

- Our Chiltern community celebrates differences and diversity within a safe and nurturing environment.
- A curriculum, which engages and enriches.
- Has ambition for all moving from novice to expert.

What does this look like within Computing?

Diversity – we ensure that all children have access to a range of programs that will build on past experiences

Engagement and enrichment – high quality lessons, teaching and resources ensure that children are able and eager to learn; immersion in experiences which have secure roots in **real-life scenarios and which build children's cultural capital**

Ambition – we want children to become critical thinkers, who can adapt their skills to meet new challenges now and in later life

At Chiltern, we want children to understand and master technology of all sorts and solve problems in almost any discipline.

Computing Skills for ALL children at Chiltern

To have accessed a wide range of software, in order that skills can be transferred and embedded

To have accessed problem-solving opportunities to build resilience and critical thinking skills

To have confidence and the skills to tackle unfamiliar technologies, based on prior experiences

To understand that these critical thinking skills are readily transferable across all subjects and situations

What we want a Computer Scientist at Chiltern to achieve

At Chiltern, we want to ensure that all children have access to a range of computing hardware and software and we make no assumptions that children have full access to this at home. We seek to ensure that computing is seen as part of their lives for work, learning and play and that the skills we have mapped out prepare them for the next stages in their education. As a result, we want all of our children to be computational thinkers. Computing teaches you how to tackle large problems by breaking them down into a sequence of smaller, more manageable problems. It allows you to tackle complex problems in efficient ways that operate at huge scale. It involves creating models of the real world with a suitable level of abstraction, and focus on the most pertinent aspects. It helps you go from specific solutions to general ones.

In short, we want children to understand and master technology of all sorts and solve problems in almost any discipline.

What we want a Computer Scientist at Chiltern to achieve

By the time they leave Chiltern, children will have had access to a quality Computing curriculum. The objectives within each strand support the development of learning across the key stages, ensuring a solid grounding for future learning and beyond. This curriculum will enable children to become logical thinkers, resilient learners and problem solvers.

Collectively, we will all strive to work together to provide a high-quality education for all, within our safe, welcoming, respectful and inclusive environment by ensuring that everyone is known and valued, with all needs being acknowledged, supported and met.

This will set in motion the development of life-long learning; the building of healthy, positive relationships and be confident in establishing broader hobbies and interests in the wider by:

- Embedding a love of learning for all futures through carefully planned opportunities across the years that encourage excellence in every subject.
- Weaving in prior knowledge to all new learning and awe and wonder hooks to inspire and enhance outcomes further.
- Creating and fostering a sense of belonging for all of our pupils and families no matter what their need, ability, culture or life experiences to date.
- Building up skills and knowledge for all, day by day, week by week, year by year, no matter what a child's starting points.

Intent

We offer a structured sequence of lessons, helping teachers to ensure that they have covered the skills required to meet the aims of the national curriculum. The spiral nature of the curriculum allows themes to be revisited at least yearly through new units that consolidate and build on prior learning. The style of curriculum design reduces the amount of knowledge lost through forgetting and ensures connections are made. The content allows for a broad, deep understanding of computing and how it links to children's lives. It offers a range of opportunities for consolidation, challenge and variety. This allows children to apply the fundamental principles and concepts of computer science. They develop analytical problem-solving skills and learn to evaluate and apply information technology. It also enables them to become responsible, competent, confident and creative users of information technology. The Computing curriculum complements and enhances a range of other subjects, such as Art, Science and Mathematics through skills learnt and vocabulary taught. This curriculum acknowledges that physical computing plays an important role in modern pedagogical approaches in computing, both as a tool to engage pupils and as a strategy to develop pupils' understanding in more creative ways. Additionally, physical computing supports and engages a diverse range of pupils in tangible and challenging tasks.

Implementation

The Computing curriculum has been written to support all pupils. Each lesson is sequenced so that it builds on the learning from the previous lesson and, where appropriate, activities are scaffolded so that all pupils can succeed and thrive. Scaffolded activities provide pupils with extra resources, such as visual prompts, to reach the same learning objectives as the rest of the class. Exploratory tasks foster a deeper understanding of a concept, encouraging pupils to apply their learning in different contexts and make connections with other learning experiences.

As well as scaffolded activities, embedded within the lessons are a range of pedagogical strategies which support making computing topics more accessible. The 12 principles embodied through the curriculum 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.

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, storytelling around concepts and finding examples of the concepts in pupils' lives.

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.

Work together – Encourage collaboration, specifically using pair programming and peer instruction as well as structured group tasks. Working together stimulates classroom dialogue, articulation of concepts and development of shared understanding.

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*.

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 use or function and evaluate it against a set of criteria.

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.

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.

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.

Add variety - Provide activities with different levels of direction, scaffolding, and support that promote active learning, ranging from highly structured to more exploratory tasks. Adapting your instruction to suit different objectives will help keep all pupils engaged and encourage greater independence.

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.

Teachers will use their professional judgement to review, select and apply relevant strategies for their pupils.

Impact

Learning in computing will be enjoyed across the school. Teachers will have high expectations and quality evidence will be presented in a variety of forms. Children will use digital and technological vocabulary accurately, alongside a progression in their technical skills. They will be confident using a range of hardware and software and will produce high-quality purposeful products. Children will see the digital world as part of their world, extending beyond school, and understand that they have choices to make. They will be confident and respectful digital citizens going on to lead happy and healthy digital lives. The reinforcement of vocabulary year-on-year will allow children to take this knowledge with them to secondary school and beyond.

The National Curriculum for Computing in England was introduced by the Department of Education in 2014. 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 curriculum can be broken down into 3 strands: computer science, information technology and digital literacy, with the aims of the curriculum reflecting this distinction.

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)

The curriculum will teach children key knowledge about how computers and computer systems work, and how they are designed and programmed. Learners will have the opportunity to gain an understanding of computational systems of all kinds, whether or not they include computers.

Key Stage 1 National Curriculum Expectations

Pupils should be taught to:

- 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.

Key Stage 2 National Curriculum Expectations

Pupils should be taught to:

- 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 concerns about content and contact.

	Computing systems and networks ¹	Creating media	Programming A	Data and information	Creating media	Programming B
Year 1	Technology around us (1.1)*	Digital painting (1.2)	Moving a robot (1.3)	Grouping data (1.4)	Digital writing (1.5)	Programming animations (1.6)
Year 2	Information technology around us (2.1)	Digital photography (2.2)	Robot algorithms (2.3)	Pictograms (2.4)	Making music (2.5)	Programming quizzes (2.6)

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 3	Connecting computers (3.1)	Stop-frame animation (3.2)	Sequencing sounds (3.3)	Branching databases (3.4)	Desktop publishing (3.5)	Events and actions in programs (3.6)
Year 4	The internet (4.1)	Audio editing (4.2)	Repetition in shapes (4.3)	Data logging (4.4)	Photo editing (4.5)	Repetition in games (4.6)
Year 5	Sharing information (5.1)	Video editing (5.2)	Selection in physical computing (5.3)	Flat-file databases (5.4)	Vector drawing (5.5)	Selection in quizzes (5.6)
Year 6	Internet communication (6.1)	Webpage creation (6.2)	Variables in games (6.3)	Introduction to spreadsheets (6.4)	3D modelling (6.5)	Sensing (6.6)

Unit summaries

	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 1	<p>Technology around us</p> <p>Recognising technology in school and using it responsibly.</p>	<p>Digital painting</p> <p>Choosing appropriate tools in a program to create art, and making comparisons with working non-digitally.</p>	<p>Moving a robot</p> <p>Writing short algorithms and programs for floor robots, and predicting program outcomes.</p>	<p>Grouping data</p> <p>Exploring object labels, then using them to sort and group objects by properties.</p>	<p>Digital writing</p> <p>Using a computer to create and format text, before comparing to writing non-digitally.</p>	<p>Programming animations</p> <p>Designing and programming the movement of a character on screen to tell stories.</p>
Year 2	<p>Information technology around us</p> <p>Identifying IT and how its responsible use improves our world in school and beyond.</p>	<p>Digital photography</p> <p>Capturing and changing digital photographs for different purposes.</p>	<p>Robot algorithms</p> <p>Creating and debugging programs, and using logical reasoning to make predictions.</p>	<p>Pictograms</p> <p>Collecting data in tally charts and using attributes to organise and present data on a computer.</p>	<p>Making music</p> <p>Using a computer as a tool to explore rhythms and melodies, before creating a musical composition.</p>	<p>Programming quizzes</p> <p>Designing algorithms and programs that use events to trigger sequences of code to make an interactive quiz.</p>



	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 3	<p>Connecting computers</p> <p>Identifying that digital devices have inputs, processes, and outputs, and how devices can be connected to make networks.</p>	<p>Stop-frame animation</p> <p>Capturing and editing digital still images to produce a stop-frame animation that tells a story.</p>	<p>Sequencing sounds</p> <p>Creating sequences in a block-based programming language to make music.</p>	<p>Branching databases</p> <p>Building and using branching databases to group objects using yes/no questions.</p>	<p>Desktop publishing</p> <p>Creating documents by modifying text, images, and page layouts for a specified purpose.</p>	<p>Events and actions in programs</p> <p>Writing algorithms and programs that use a range of events to trigger sequences of actions.</p>
Year 4	<p>The internet</p> <p>Recognising the internet as a network of networks including the WWW, and why we should evaluate online content.</p>	<p>Audio editing</p> <p>Capturing and editing audio to produce a podcast, ensuring that copyright is considered.</p>	<p>Repetition in shapes</p> <p>Using a text-based programming language to explore count-controlled loops when drawing shapes.</p>	<p>Data logging</p> <p>Recognising how and why data is collected over time, before using data loggers to carry out an investigation.</p>	<p>Photo editing</p> <p>Manipulating digital images, and reflecting on the impact of changes and whether the required purpose is fulfilled.</p>	<p>Repetition in games</p> <p>Using a block-based programming language to explore count-controlled and infinite loops when creating a game.</p>



	Computing systems and networks	Creating media	Programming A	Data and information	Creating media	Programming B
Year 5	<p>Sharing information Identifying and exploring how information is shared between digital systems.</p>	<p>Video editing Planning, capturing, and editing video to produce a short film.</p>	<p>Selection in physical computing Exploring conditions and selection using a programmable microcontroller.</p>	<p>Flat-file databases Using a database to order data and create charts to answer questions.</p>	<p>Vector drawing Creating images in a drawing program by using layers and groups of objects.</p>	<p>Selection in quizzes Exploring selection in programming to design and code an interactive quiz.</p>
Year 6	<p>Internet communication Recognising how the WWW can be used to communicate and be searched to find information.</p>	<p>Webpage creation Designing and creating webpages, giving consideration to copyright, aesthetics, and navigation.</p>	<p>Variables in games Exploring variables when designing and coding a game.</p>	<p>Introduction to spreadsheets Answering questions by using spreadsheets to organise and calculate data.</p>	<p>3D modelling Planning, developing, and evaluating 3D computer models of physical objects.</p>	<p>Sensing Designing and coding a project that captures inputs from a physical device.</p>



National Curriculum Coverage – Key Stage 1 Computing Curriculum	1.1 Technology around us	1.2 Digital painting	1.3 Moving a robot	1.4 Grouping data	1.5 Digital writing	1.6 Programming animations	2.1 Information technology around us	2.2 Digital photography	2.3 Robot algorithms	2.4 Pictograms	2.5 Making music	2.6 Programming quizzes
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	✓				✓	✓	✓			✓		



National Curriculum Coverage – Years 3 and 4

	3.1 Connecting computers	3.2 Stop-frame animation	3.3 Sequencing sounds	3.4 Branching databases	3.5 Desktop publishing	3.6 Events and actions in programs	4.1 The Internet	4.2 Audio editing	4.3 Repetition in shapes	4.4 Data logging	4.5 Photo editing	4.6 Repetition in games
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 concerns about content and contact							✓	✓			✓	



National Curriculum Coverage – Years 5 and 6	5.1 Sharing information	5.2 Video editing	5.3 Selection in physical computing	5.4 Flat-file databases	5.5 Vector drawing	5.6 Selection in quizzes	6.1 Internet communication	6.2 Webpage creation	6.3 Variables in games	6.4 Introduction to spreadsheets	6.5 3D modelling	6.6 Sensing
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 concerns about content and contact	✓	✓						✓	✓		✓	



All learning objectives have been mapped to the taxonomy of ten strands, which ensures that units build from one Key Stage to the next. Within the Computing curriculum, every year group learns through units within the same four themes, which combine the ten strands. This approach allows us to use the spiral curriculum approach to progress skills and concepts from one year group to the next.

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