



COMPUTING CURRICULUM OVERVIEW FOR GOVERNORS

OUR SCHOOL VISION

“Striving for excellence together in a caring Christian community.”

RESPECT COMPASSION COURAGE

As a Church school, we believe that people grow in mind, body and spirit. Christian values are the foundation of our teaching and our ethos as we strive together for excellence for all. We aim for each member of our school community to fully engage in the great adventure that is Primary education.

Working together, we aim for all of our school community to become:

- successful learners who enjoy learning and exploration, make progress and achieve;
- confident, well-rounded individuals who are able to live safe, healthy and fulfilling lives; and
- responsible citizens with strong moral and social values who make a positive contribution to society.

“I came to give life—life in all its fullness.” John 10:10

LIVING OUR VISION THROUGH COMPUTING

Our Computing curriculum helps children become confident, creative and caring digital citizens who work together to solve real problems and improve their communities. Children collaborate on coding projects, group enquiries and multimedia tasks that build teamwork, curiosity and resilience while reflecting our school values of Respect, Compassion and Courage. Through computing they explore digital systems and emerging technologies, learn to make safe choices online and use their skills to contribute positively to school life and beyond. Learning is rooted in real-world contexts so children see how their ideas can have a positive impact on people and the planet.

YEAR GROUP BREAKDOWN (TERM-BY-TERM)

Each year shows the three Teach Computing units taught across the year, with the primary focus for the term and the key content and pupil skills teachers should plan to evidence.

Year 1

Term	Unit (focus)	Key content and skills
Autumn	Technology around us (Computing systems, algorithms)	Identify technology in school; vocabulary (device, computer); basic mouse skills; safe use – where to get help online.
Spring	Digital painting (Creating media)	Use paint tools; experiment with shape, colour and brush size; compare digital and paper art; save work.
Summer	Moving a robot (Algorithms, programming)	Give precise instructions; sequence and predict outcomes with floor robots; plan simple routes.

Year 2

Term	Unit (focus)	Key content and skills
Autumn	Information technology around us (Networks, systems)	Recognise IT in different contexts; how devices can work together; discuss responsible use.



Term	Unit (focus)	Key content and skills
Spring	Pictograms (Data & information)	Collect data, create tally charts and pictograms using software; answer simple questions from data.
Summer	Programming animations (Programming, design)	Use ScratchJr/Scratch concepts; join blocks; create simple animations; follow and test algorithms.

Year 3

Term	Unit (focus)	Key content and skills
Autumn	Connecting computers (Networks, systems)	IPO (input/process/output); identify parts of a network; benefits of connected devices.
Spring	Sequencing sounds (Programming, creating media)	Introduce Scratch; build sequences combining motion and sound; design → code → test.
Summer	Events and actions (Programming, design)	Events causing actions; maze movement; introduction to extensions (pen), debugging.

Year 4

Term	Unit (focus)	Key content and skills
Autumn	The internet (Networks, safety & security)	Internet = network of networks; WWW vs internet; evaluating online content; copyright and trust.
Spring	Data logging (Computing systems, data & information)	Sensors and data loggers; collect and analyse time-series data; pose and answer questions with data.
Summer	Repetition in games (Programming, design)	Loops (count-controlled, infinite); design and create Scratch games using repetition and concurrency.

Year 5

Term	Unit (focus)	Key content and skills
Autumn	Systems and searching (Networks, effective tools)	Systems thinking (inputs/processes/outputs); how search engines index and rank; refine searches critically.
Spring	Vector drawing (Creating media, tools)	Create vector-style graphics (Google Drawings); layers, grouping, alignment; prepare assets for publishing.
Summer	MakeCode & micro:bit (Programming, sensing)	Introduce physical computing with micro:bit; simple inputs/outputs, real-world sensing and program design.

Year 6

Term	Unit (focus)	Key content and skills
Autumn	Internet communications (Networks, safety & tools)	Deeper study of data transfer and secure communication; evaluating sources and digital citizenship.
Spring	Introduction to spreadsheets (Tools, data & information)	Spreadsheets for analysis: formulas, charts, modelling and data-driven decisions.
Summer	Sensing (micro:bit) (Programming, systems)	Advanced micro:bit projects integrating sensors, data logging and decision structures.



PROGRESSION

Year	Computing & Networking Systems	Programming	Digital Creativity
Year 1	Recognise everyday technology; name basic device parts; understand devices help people; know who to ask for online help.	Give precise, ordered commands to move floor robots; follow simple sequences; predict outcomes.	Use simple paint tools; experiment with colour/shape; save/reopen work; compare digital and paper art.
Year 2	Identify IT in school and community; describe simple device interactions (e.g., scanner → till); understand responsible use.	Join short command blocks (ScratchJr); create simple animations; follow and test basic algorithms; fix obvious errors.	Collect simple data (tally); create pictograms using software; read and answer questions from digital charts.
Year 3	Explain input → process → output; identify basic network components; describe benefits of connected devices.	Build longer sequences in Scratch; use events to start actions; follow design → code → test cycle; start debugging.	Combine sound and media in projects; use drawing/painting apps for structured tasks; begin saving digital evidence.
Year 4	Describe the internet as a network of networks; distinguish WWW vs internet; evaluate reliability and ownership of online content.	Use loops and concurrent actions; read and modify existing code; apply systematic debugging strategies.	Use data loggers/sensors to collect time-series data; analyse and present data; create richer multimedia projects.
Year 5	Analyse systems at small and larger scale; explain information flow between systems; understand search indexing and ranking.	Design and implement multi-stage programs; use micro:bit/physical computing basics; apply abstraction and modularity.	Produce vector-style graphics; manage layers/groups; prepare digital assets; source and evaluate digital content critically.
Year 6	Understand secure data transfer and basic privacy concepts; apply systems thinking to real problems; critique digital services.	Combine inputs, conditionals and data handling; integrate sensing and control; plan, test and refine substantial programs.	Use spreadsheets for modelling and analysis; create polished digital artefacts combining media and code; present outcomes for an audience.

ENRICHMENT OPPORTUNITIES

- STEM club micro:bit projects; whole-school digital creativity showcase; data-driven science links; Internet Safety Week assemblies and children's panels; computing ambassadors to link learning with real-world practice.

TEACHING AND LEARNING APPROACHES

- Teach iterative design: design → implement → test → refine.
- Use live modelling for coding and scaffold with levels of abstraction (Task → Design → Code → Run).
- Use unplugged activities to build conceptual understanding before on-screen work.



- Formative checks: mini-quizzes, live testing, children's explanations and child-friendly success criteria.

Support for SEND and EAL

- Use visual algorithms, step templates, sentence stems, word banks and BSL where required.
- Pair programming and targeted TA support.
- Offer alternative input methods and differentiated success criteria.

Home-School Partnership

- Parent activities: supervised micro:bit starters; talk through how to spot unreliable content; share saved children's projects.
- Communication: ParentHub for parent messages and whole-school updates; Google Classroom to share resources and information with children.

ASSESSMENT, POSSIBLE EVIDENCE FOR MONITORING AND STATUTORY INFORMATION

Assessment approach

- Formative: lesson success criteria, live demonstrations, children's self/peer assessment (thumbs up/sideways/down).
- Summative: unit rubrics (assessment guides), saved digital portfolios (project files/screenshots), Sonar unit outcomes.

Possible evidence for monitoring

- One exemplar saved project per year group.
- Short children's reflection for KS2 (1–2 sentences about what they did/learned).
- Photos and floor books for EYFS/KS1 to capture discussion-led learning.
- Sonar entries summarising unit outcomes and next steps.
- Screenshots, exported project files or links to children's work (where accounts permit).

Statutory coverage

- The full National Curriculum for Computing is covered across Y1–Y6 (programming, computing systems, networks, data and digital literacy).
- Online safety is taught throughout and reinforced during Internet Safety Week.

GLOSSARY OF TECHNICAL TERMS

- Algorithm: an ordered set of instructions to achieve a task (in computing this becomes a program).
- Rubric: an assessment guide that explains the criteria used to judge children's work.
- Sonar: the school's assessment/recording system used to log unit outcomes and progress.
- Micro:bit: a small programmable device used for physical computing and sensing projects.
- Data logger / sensor: hardware that records measurements (temperature, light, sound) over time for analysis.
- Vector graphics: images made from shapes and lines (scalable without losing quality).
- AI literacy: age-appropriate understanding of how artificial intelligence can influence search results and content, and how to evaluate AI outputs critically.
- Scratch / ScratchJr / MakeCode: block-based programming environments used at different ages to teach coding concepts.