St Michael's CE Primary School, Sandhurst
Subject Vision Document: Computing

Vision (Intent)

At St Michael's CE Primary School the aims in Computing are:

- To provide a broad, balanced, challenging and enjoyable Computing curriculum for all pupils
- To develop pupils' computational thinking skills, to benefit them throughout their lives
- To meet the requirements of the National Curriculum programmes of Study for computing at Key Stage 1 and 2
- To respond to new developments in technology
- To equip pupils with the confidence and skills to use digital tools and technologies throughout their lives
- To enhance and enrich learning in other areas of the curriculum using computing
- To develop the understanding of how to use computers and digital tools safely and responsibly.

Curriculum (Implementation)

We use the Teach Computing Curriculum. This is a comprehensive collection of materials produced to facilitate the delivery of the computing curriculum. The Teach Computing Curriculum was created by the Raspberry Pi Foundation on behalf of the National Centre for Computing Education (NCCE). As all the resources are editable under the Open Government Licence (OGL), we can tailor the resources to our school (and individual teacher) setting.

Curriculum Structure

The Teach Computing Curriculum is structured in units. For these units to be coherent, the lessons within a unit must be taught in order. However, across a year group, the units themselves do not need to be taught in order, with the exception of 'Programming' units, where concepts and skills rely on prior learning and experiences. The units for both key stages 1 and 2 are based on 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.

Our Computing curriculum uses the National Centre for Computing Education's computing taxonomy to ensure comprehensive coverage of the subject. This has been developed through a thorough review of the computing programme of study. All learning outcomes can be described through a high-level taxonomy of ten strands, ordered alphabetically as follows:

Algorithms — Be able to comprehend, design, create, and evaluate algorithms

St Michael's CE Primary School, Sandhurst
Subject Vision Document: Computing

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

The scheme 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.

Lesson Delivery (Pedagogy)

Computing is a broad discipline, and computing teachers require a range of strategies to deliver effective lessons to their pupils. The National Centre for Computing Education's pedagogical approach consists of 12 key principles underpinned by research: each principle has been shown to contribute to effective teaching and learning in computing.

These 12 principles are embodied by the Teach Computing Curriculum, and examples of their application can be found throughout the units of work at every key stage. St Michael's teachers use their professional judgement to review, select, and apply relevant strategies for their pupils:

- **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.
- **Structure lessons** —Use supportive frameworks to ensure that differentiation can be built in at various stages of the lesson.
- **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 and around the school
- **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' (ncce.io/qr06), can help pupils develop a secure understanding of complex concepts.
- **Work together** —Encourage collaboration, specifically using pair working, peer instruction and 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 user or function, and evaluate it against a set of criteria.
- **Model everything** — Model processes or practices 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, especially debugging.

Assessment

Every lesson in the Teach Computing scheme includes formative assessment opportunities for teachers to use. These opportunities are listed in the lesson plan and are included to ensure that misconceptions are recognised and addressed if they occur. The learning objective and success criteria are introduced in the slides at the beginning of every lesson. At the end of every lesson, pupils are invited to assess how well they feel they have met the learning objective using thumbs up, thumbs sideways, or thumbs down. This gives pupils a reminder of the content that has been covered, as well as a chance to reflect. It is also a chance for teachers to see how confident the class is feeling so that they can make changes to subsequent lessons accordingly.

Every unit includes a summative assessment in the form of either a multiple choice quiz (MCQ) or a rubric. All units are designed to cover both skills and concepts from across the computing national curriculum. Units that focus more on conceptual development include an MCQ. Units that focus more on skills development end with a project and include a rubric. However, within the 'Programming' units, the assessment framework (MCQ or rubric) has been selected on a best-fit basis.

St Michael's CE Primary School, Sandhurst

Subject Vision Document: Computing

Curriculum Enhancement

An after-school Code Club is available for UKS2 children with an especial interest in coding.