Subject Policy
Computing

Reviewed by: Heather Rowe
Date: September 2023
Review Date: September 2025

INTENT

At Debden Church of England Primary Academy, all of our curriculum disciplines are used to underpin our school vision, which is to ensure that the children in our care:

- Progress exceptionally well academically, across a broad and knowledge-rich curriculum;
- Develop into confident compassionate, well-rounded individuals, in a safe, caring, Christian environment;
- Become equipped with the learning skills needed to deal with future challenges;
- Create happy, positive memories of their childhood.

The National Curriculum Purpose of Study states that:

"A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world."

Our computing curriculum has been carefully constructed through close consideration of both the expectations of the National Curriculum and the vision and contextual requirements of our school and its children. Our aim is to produce learners who are confident, discerning and effective users of technology and who also have a good understanding of computers and how computer systems work, and how they are designed and programmed+.

We strive to achieve this aim by:

- Supporting all children in using technology with purpose and enjoyment;
- Meeting, and building on the minimum requirement set out in the National Curriculum as fully as possible and helping all children to achieve the highest possible standards of achievement;
- Helping all children to develop the underlying skills and capability which is essential to developing Computing capability (such as problem solving, perseverance, learning from mistakes) and apply them elsewhere;
- Helping all children to develop the necessary skills to exploit the potential of technology and to become autonomous and discerning users;
- Helping all children to evaluate the benefits and risks of technology, its impact on society and how to manage their use of it safely and respectfully.

Our curriculum is built upon and adapted from the principles and schemes of the National Centre for Computing Education's 'Teach Computing' curriculum. Created by subject experts, using the latest pedagogical research and teacher feedback, these resources are built around an innovative progression framework where computing content has been organised into interconnected networks. Children's knowledge and skills are progressed through the following units (which are developed upon year-on-year of their primary education);

- Computing Systems and Networks;
- Creating Media;
- Programming;
- Data and Information;
- E-Safety.

We have interwoven the use of a range of computer programs and software throughout the units, to ensure that children leave primary school highly familiar with the computer systems and programs that they will need throughout their lives (e.g. Microsoft Word, Publisher, Google Forms, etc.) We have also developed knowledge organisers for each unit, and a progression of skills map to plot an ambitious rate of development for the children throughout their time at Debden.

Both plugged and unplugged learning opportunities are planned to support pupils' understanding of the underlying concepts in Computing. These opportunities may well be presented within other subject areas (e.g. sequencing instructions in English, problems solving in Maths or isolating variables in Science).

IMPLEMENTATION:

Computing is taught weekly for a period of 45 minutes to 1 hour per week (increasing with age). Computing sessions across the school are staggered over the week to ensure that all children have access to the digital resources that they need – we have invested significantly over the past couple of years to ensure that there are enough ipads and laptops in the school, for example, to ensure that each individual child in a class can access a device each throughout their computing session.

As with all subjects, teachers apply our agreed Debden pedagogical strategies, including:

1. Using the 'Principles of Instruction,' (e.g. regular review, small steps material) 'Teach Like a Champion' (e.g. cold calling, talk partners) and other research-approved pedagogical strategies to maximise children's engagement and progress and to continuously assess understanding.
2. Taking every opportunity to develop children's reading skills and vocabulary, through the use of a range of model texts and vocabulary-building strategies (which are displayed in every classroom);
3. Incorporating daily, weekly and monthly recall activities in order to ensure that children's understanding of important concepts is remembered and strengthened.
4. Encouraging children to develop their learning skills, such as resilience and independence, through models, guided learning practice, effective support and self-help strategies (see right).
5. Tailoring learning to the needs of the children, adapting plans in the moment if necessary, based on continuous and accurate assessment for learning.

In addition to this, in Computing, all teachers aim to adopt the 12 pedagogy principles of Computing (see appendix A).

IMPACT:

Outcomes on digital files and in floor books – Children have their own digital space on within the pupil files section of our school network. Teachers regularly access and review this work in order to form their feedback and inform future planning. To enable the class to follow their collective learning journey and to celebrate their successes, the class teacher also maintains a 'computing floor book', which showcases a number of the learning highlights from across each teaching unit.

Whole-Class Feedback – In addition to the verbal feedback that is provided within each computing lesson, teachers are expected to complete at least one whole-class feedback cycle within each computing block. This includes highlighting strengths, identifying misconceptions and next steps, and commenting upon the children's presentation. A 'blue sticker task' (the call to action) should be utilised to address misconceptions and learning that has not been secured.

End of Unit Assessment – -At the end of each unit, children are awarded a score (out of 15) for:

1.Their skill level in the particular area of computing being studied (out of 10 marks: using the skills map). They are marked using the following criteria:

Score (0-10 Scale)	Descriptor
0	Does not demonstrate any of the skills in the appropriate section/s of the skills map.
1-2	Demonstrates a small minority of the skills in the appropriate section/s of the skills map.
3-4	Demonstrates an increasing number of the skills in the appropriate section/s of the skills map.
5-6	Demonstrates around half of the skills in the appropriate section/s of the skills map.
7-8	Demonstrates most of the skills in the appropriate section/s of the skills map.
9-10	Demonstrates all of the skills in the appropriate section/s of the skills map.

2. Their knowledge of the concepts in the unit that has been studied (out of 5 marks: using a Debden Foundation Assessment Paper). The paper should include 2 Remembering questions, 2 Understanding questions, and 1 deeper level question. These assessment papers are overseen by the subject leader, for consistency, and check children's understanding of the key concepts set out in the knowledge organisers.

Monitoring – Twice per year, subject monitoring of Computing takes place, which is normally carried out by the subject leader (at times, this may be a senior leader). At least once per year, this is carried out alongside the link Governor for the subject. The subject monitoring process includes:

-Lesson visits;

-Book look;

-Student chats;

-Checking of student understanding of information on knowledge organisers;

-Viewing classroom displays;

-Conversations with teachers;

-Analysis of assessment data;

-Subject leader 'deep-dive' questions and review of key subject documents (when with link Governor).

The information gathered from teachers is fed back in a timely fashion via our subject leader monitoring reports.

REVIEW:

This policy will be regularly reviewed by the Subject Leader and the Head of School. The maximum period between reviews is a period of two years.

APPENDIX A – Pedagogy Principles

National Centre for Computing Education

Raspberry Pi

How we teach computing

12 pedagogy principles

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.

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 (semantic waves) can help pupils develop a secure understanding of complex concepts.

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.

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.

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.

Work together

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

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.

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.

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.

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.

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.

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.

Find out more about our principles and add some or all to your personal pedagogy toolkit.

ncce.io/pedagogy