

Computing Teaching and Learning Guide

Intent

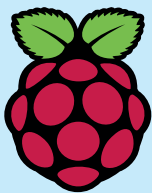
The Forest of Dean displays average levels of deprivation on the whole, however Cinderford (a town in the catchment area) ranks in the top 20% most deprived places in the country. In the Forest of Dean 17% of the Authority's population live in rural areas and are under 19 years of age. Consequently there still exists a digital divide with a number of students having limited access to digital devices at home or suitable internet speeds due to their remote rural locations.

Taking this into account it is essential that our curriculum provides opportunities for all to develop core computer literacy skills within the school environment through lessons and wider opportunities. We aim to give students essential knowledge and skills at each key stage that will prepare them for the next stage of study or the workplace. With an aim to help students become effective workers in a digital future, we guide students to become progressively more independent and develop a range of life skills:

- problem solving and critical thinking
- logical thinking
- communication and collaboration
- creativity
- self-evaluation

At Drybrook Primary School we follow the Teach Computing scheme which focuses on a progression of skills in digital literacy, computer science, information technology and online safety to ensure that students become competent in safely using, as well as understanding, technology.

The work of the NCCE is underpinned by our 12 principles of Computing Pedagogy, below. They are also available to download as a poster [12 pedagogy principles](#).



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.

nccce.io/pedagogy

Implementation

Teaching is designed to help children to remember in the long term the content they have been taught and to integrate new knowledge into larger computing concepts.

These will be based on research into the best way to deliver learning of that specific subject. Use of EEF, Ofsted research and other research will be drawn on to provide a high-quality framework for lesson plans to be built around.

Timetabling

All subjects are taught discretely to ensure each subject has its own disciplinary structure to be respected and substantive knowledge to be learned and approach the clarity of the curriculum goals so they are clear to all.

Individual lesson structure

Recap prior learning both long term and short term – retrieval practice

- New learning begins from a previously comfortable starting point – link to prior lesson, country, event etc.

Evidence base

Rosenshine's principles of instruction

- *Begin a lesson with a short review of previous learning.*
- *Present new material in small steps with student practice after each step*
- *Engage students in weekly and monthly review.*

Medium term plans structure

Key foundational knowledge to be retrieved over time through retrieval practice

- Use of a knowledge organiser to refer back to key foundational knowledge.