

# Planning guidance



## How do you plan a scheme of work?

It's sensible to establish a clear overall structure to your computing planning. This will help teachers and pupils find their way around the new curriculum and give some sense of continuity and progression to your planning. The national curriculum themes of computer science, information technology and digital literacy (foundations, applications and implications) are a good way to do this. Within each of these three themes you should aim to develop units of work that provide:

- a broad and balanced coverage of computing for each year group
- clear progression of understanding, knowledge and skills from one year to the next.

Many of your units might take the form of creative projects, in which pupils master the ideas of the curriculum through meaningful, practical activities. Once you have a set of units, map these back to the content of the programmes of study. Check that, taken as a whole, your units adequately cover all the content. If they don't, make some changes.

Don't see the curriculum as a limit to what you can teach. Some schools include units on computer hardware, touch-typing, cryptography or developing and designing content for the Web in HTML.

A good scheme of work should be continually developed as you assess pupils' learning and both you and they evaluate your teaching. This can be thought of as part of a continuous cycle of iterative development – 'debugging' units of work and looking at how new features can be implemented.

### Adapting your existing content

In the transition from ICT to computing, the emphasis has shifted from skills in finding and presenting information to understanding the processes and applications of computation. Even so, don't think you have to throw away your old scheme of work. The best units are very likely to be relevant to the new computing curriculum, although you'll probably find you need to make room for new activities that:

- develop pupils' programming skills
- focus on computational thinking
- develop pupils' understanding of networks including the internet.

### Adopting published schemes of work

Don't feel that you have to write your own scheme of work for computing. There are plenty of published schemes already available. Examples include:

- **commercial schemes:** e.g. *Switched on Computing* (Rising Stars), *100 Computing Lessons* (Scholastic) and *Fantastict Computing Framework* (Fantastict).
- **local authority schemes:** e.g. Bury, 'Bury Primary Computing Solution'; Somerset, 'A Computing Curriculum for Primary Learners' and Cambridgeshire, 'The Cambridgeshire Progression in Computing Capability Materials' (see Further resources).
- **schemes by individual teachers:** some of which have been shared freely through the CAS Community site: e.g. those by Phil Bagge, Matthew Wimpenny-Smith and Jon Chippindall (see Further resources).

You can choose to adopt or adapt these, taking into account the resources available, the interests and enthusiasms of you, your colleagues and pupils, and your school's approach to cross-curricular or integrated topics.

## Guidance from Ofsted

In the past, Ofsted has provided detailed guidance on inspecting a school's ICT curriculum. The lead HMI for computing and e-safety, David Brown, gave some indications of his views on a good or outstanding computing curriculum in a presentation in July 2014 (see Further resources). It's well worth considering his expectations when planning or adapting a scheme of work. He suggested the following:

- *the curriculum is broad and balanced with all three computing strands covered well for all pupils, in computing lessons and/or across the school curriculum*
- *the contexts in which computing is taught are relevant to pupils' lives and reflect the increasing use of computing in the world of industry*
- *pupils are expected to use their computing knowledge, skills and understanding in realistic and challenging situations*
- *pupils have comprehensive knowledge and understanding of how to stay safe when using new technologies*
- *rigorous curriculum planning ensures the subject makes an outstanding contribution to pupils' spiritual, moral, social and cultural development.*<sup>1</sup>

### How should we plan a computing lesson?

Don't assume that the conventional 'three-part' lesson must be followed. Often in computing pupils will be learning through making things and it's helpful to give longer periods of time for such projects. Not all lessons need to be about using a particular bit of software to do something. Even where pupils are learning to use new software, it can be worth starting with a discovery phase. They can then share their discoveries with one another, but make sure you correct any misconceptions.

## Unplugged activities

Elements of the curriculum, particularly logical reasoning, algorithms, decomposition or how networks work, can be addressed without a computer, using pencil and paper or role-play activities. Both Barefoot Computing and CS Unplugged (see Further resources) have excellent activities for developing understanding rather than just practising coding skills.

Computational thinking should be embedded through the computing curriculum (see pages 6–17).

<sup>1</sup> Brown, D., Ofsted National Lead for Computing, 'Inspecting computing' slides (Barefoot Computing Conference).

## Differentiation

Prior learning can vary quite widely across a class. Collaborative group work or paired programming (or other) exercises provide opportunities for pupils to teach new things to one another.

Allow enthusiastic pupils the autonomy (and critical support) to set their own challenges. For example, if some of your pupils are already adept at coding in Scratch, set them the challenge of coding the same algorithm in another language.

Think about inclusion.

- Is the task chosen so that all pupils' enthusiasm can be engaged and you've avoided issues about which some pupils might be particularly sensitive?
- How can you make appropriate adaptations for SEN/D or EAL pupils? Assistive technology can help with SEN/D, as can using an icon-based rather than text block-based programming environment: some dyslexic pupils are likely to find Kodu more accessible than Scratch for this reason.
- How can you help pupils who do not have access to digital technology at home become more confident and independent in computing?

## www Further resources

- Barefoot Computing, teaching resources, available at: <http://barefootcas.org.uk/> (free, but registration required).
- Computing At School, community site resources, available at: <http://community.computingatschool.org.uk/resources>.
- Computing At School (CAS) Include: Computing Science for All, available at: <http://casinclude.org.uk/>.
- CS Unplugged, free activities and resources, available at: <http://csunplugged.org/>.
- ICTdotcom, ICT and Computer Planning by Matthew Wimpenny-Smith, available at: <http://mwimpennys.primaryblogger.co.uk/uncategorized/ict-and-computing-planning/>.
- Junior Computer Science, resources by Phil Bagge: [www.code-it.co.uk/](http://www.code-it.co.uk/).
- Sheffield ILS eLearning Team 'SEN Computing Professional Learning Community', available at: <http://sencomputing.wikispaces.com/>.
- Somerset County Council 'A Computing Curriculum for Primary Learners', available at: <http://bit.ly/1d1P2OK>.
- The ICT Service 'Cambridgeshire Progression in Computing Capability Materials', available at: [www.ccc-computing.org.uk/](http://www.ccc-computing.org.uk/).