Naace
Advancing education through technology

Assessment

Guidance from the Naace Assessment Panel

Assessment is an integral part of the curriculum and cannot be separated from it - hence a school approach to assessment will need to be tailored to match their approach to the curriculum. Assessment should inform next steps for pupils, whether that is formative or summative.

The nature of Computing as a subject is very practical. Hands on learning in embedded contexts which provide opportunities for the explicit development of knowledge skills and understanding can bring its own challenges for the assessment of what pupils have learnt in Computing. Sometimes it is difficult to separate out the learning in Computing from the learning that has happened in the subject that provides the context. Or it can be difficult to separate out the learning of individual pupils who have worked collaboratively on a project. However, the context for learning is incredibly important. Formative assessment should be in those contexts where Computing has a clear purpose and where knowledge, skills, understanding and critical analysis are developed explicitly and intentionally, rather than in a context where pupils are assumed to be learning by osmosis simply because they are using technology. Contexts in primary and secondary schools will differ somewhat, as opportunities for extended cross-curricular projects are often taken at primary but pose different challenges within a secondary curriculum.

Summative assessment will no longer include levels and the programme of study describes what pupils should be able to do by the end of each key stage. It is anticipated that pupils will build a portfolio of evidence of work throughout a key stage that demonstrates their learning in each of those bullet points. Some evidence may relate to more than one bullet point in the programme of study and some bullet points will have multiple pieces of evidence. Naace have been working closely with TLM in the development of more formal qualifications in Computing, and further information can be found at http://www.naace.co.uk/curriculum/qualifications.

Formative assessment, or assessment for learning, will use many of the techniques common to teaching across other areas of the curriculum and the guidance from the Naace assessment panel focuses on the practicalities of assessment, especially for areas of the subject, such as programming, where many teachers have much less experience of teaching, learning and approaches to formative assessment.

Naace
Advancing education through technology

It is not always helpful to use checklists when evaluating a piece of Computing work. Getting an overall picture of piece of work is more akin to looking at the overall impact of a piece of writing. In Computing, work that demonstrates a higher level of knowledge, skills and understanding is both effective AND efficient. There may be able pupils who find very creative solutions to the problem they are working on that do not fit the pattern expected by a class teacher. Assessing their understanding is often most effectively done through effective questioning. It is important to remember that effective teaching uses a facilitative "guide on the side" approach rather than the "sage on the stage". So as teachers who may be developing their own subject knowledge and confidence seek to recognise, assess and provide next steps for learning when they are possibly at the upper limits of their own knowledge, skills and understanding, approaches to assessment used in other areas of the curriculum are relevant for Computing, such as using self-and peer-assessment so pupils are appropriately assessed and challenged to develop their learning further.
Open-ended questioning techniques or "technical interviews" encourage pupils to explain and justify their approaches to solving a problem. Technical interviews/conferences may be between:

- a pupil and teacher
- a pupil and their peer(s)
- a pupil and an expert from outside the class (as mediated and facilitated by the teacher)

Possible questioning/discussion approaches:

- "Compare and contrast"
- What have you done?
- Why have you done it?
- Why have you chosen this way?
- What other way could you have done this?
- How could you improve what you have done?

Pupils can be taught how to give constructive and meaningful feedback - this should be modelled by the teacher, as can the culture of sharing that computing enables/benefits from. Don't be afraid to ask questions during the lesson, get children up to explain, "stop and share"/mini plenaries throughout lessons support learning and formative assessment.
Online communities, such as the Scratch or Kodu communities or other open source communities, provide a rich source of inspiration and shared products such as programs. These can be a useful starting point for pupils finding ways to achieve an end goal, and they can also be a useful place for those who need feedback from people who are knowledgeable about coding using that particular tool. Any use of such communities by teachers should be

approached in a careful and considered way that promotes safe and responsible use. For example, names of pupils/schools would need to be protected. Support forums, such as Naace Talk Lists, should be seen as a helpful resource for teachers seeking advice about challenging pupils in areas of the subject they feel less confident about.

"Mistakes" in programming are an important part the learning process, often resulting in "bugs" that mean a program doesn't do what is expected. Part of the curriculum involves developing skills in "debugging" - or finding these "mistakes" and correcting them. Breaking problems down into different parts is "decomposition", a problem solving technique that develops computational thinking. Formative assessment should not just focus on the product, such as a game that has been programmed, but on the way that debugging and decomposition have been used.

As pupils build evidence towards the end of key stage goals, it may be helpful to distinguish between the different levels of learning that are taking place within a task. Bloom's taxonomy, or Bloom's digital taxonomy, can provide a framework for evaluating these levels of learning. Other approaches include the use of "I can" statements, developed by the school or taken from a commercial scheme - in which case it is often helpful to group these statements into "emergent" and "extended" skills. Others have felt it appropriate to keep a system of "levels" that teachers are comfortable with, adapting it and adding in their own additional statements - though it is worth remembering that the government have scrapped levels and their descriptors and will not be replacing them, and have made it clear that schools must make their own decisions about formative assessment.

Naace
Advancing education through technology

Summary of points and considerations for a Computing curriculum

- Assessment is an integral part of the curriculum and cannot be separated from it - hence a school approach to assessment will need to be tailored to match their approach to the curriculum. Assessment should inform next steps for pupils, whether that is formative or summative
- Developing staff confidence and competence in teaching and assessing computing is a journey
- the National Curriculum is only part of what a school does - it then needs to be developed into a localised curriculum that is relevant for the pupils in the school. The National Curriculum is not the same as the school curriculum.
- discrete computing in an embedded context is important- not learning by osmosis, but learning computing for a purpose and which allows for explicit development of knowledge, skills and understanding, e.g. developing digital communication skills for collaboration.
- when being taught as part of another subject (more relevant in primary schools, when Computing may be developed in the context of a series of e.g. history lessons or a cross curricular project), additional time is added to series of lessons to allow for focused computing input - Fluid Timetabling
- flexible approaches to lessons ar more helpful than formulaic approaches
- teachers may not know all the answers - but effective teaching uses a facilitative “guide on the side” approach rather than the “sage on the stage”
- there is much advice and support for teachers that can support both teaching and assessment, as well as places to go to with specific questions - e.g. support forums, Naace talk lists
- mistakes in programming are an important part the learning process (debugging, decomposition, problem solving)
- use of checklists is not always helpful - often it is important to look at the overall impact of any outcome - e.g. the efficiency and effectiveness of a piece of programming may lead to a shorter piece of work than someone who has achieved the same outcome with a less efficient method, even though they may have used all the different techniques that are part of the lesson’s success criteria. It is more akin to marking a piece of writing.
- pupils should know what they need to do to get better, be able to identify the next stps in their learning
- TIBs - close relative of WALT and WILF (This is because) - is part of building understanding of computing

- digital pupil portfolios that provide ownership and access beyond the classroom are to be recommended
- technology can be used to support personalised assessment in the same way as it is able to support learning in other subjects
- it is important to include and acknowledge learning that takes place outside school and to be aware of pupils' skills and activities beyond the classroom
- How do teachers recognise and assess what is possibly at the upper limits/beyond their own knowledge, skills and understanding? Technical interviews/conferences, peer feedback, feedback from external experts, portfolio annotation
- "I can" statements need to be flexible and adaptable so new ones can be added as new technologies are being used in school, to meet the needs of a local school curriculum
- "I can" statements can usefully be grouped - e.g. emergent/extended
- pupils can be taught how to give constructive and meaningful feedback - this should be modelled by the teacher, as can the culture of sharing that computing enables/benefits from
- Don't be afraid to ask questions during the lesson, get children up to explain, "stop and share"/mini plenaries throughout lessons support learning and assessment
- bought-in schemes of work may be useful in schools as a basis for their own Computing curriculum - especially at a time of transition, if appropriate continuing professional development is in place to enable staff to continually review and update the school curriculum. Schools should be open to adapt any bought-in scheme to localise it - they need to take ownership of their curriculum, look for a range of software/options about devices/tools; any curriculum needs to be flexible to meet the needs of the children in the class.
- developing staff confidence and competence can be supported by technical support at crucial points as they start to use new tools/techniques, e.g. team teaching, floor walkers providing technical support
- there are a range of approaches possible for Computing that include: school-developed levels, "I can" statements, Bloom's taxonomy

Bloom's taxonomy:

- seems to be most appropriate for KS2/3/4
- well-suited for assessing open-ended/project based learning
- whilst Bloom' taxonomy is hierarchical, this will need to be treated in a less linear way if being used to describe a pupils' learning
- Harrow and Simpson's approach to assessing learners who are acquiring the skills for physically doing something can be used to further distinguish between the learning that

is going on at the Knowledge/Comprehension/Application stages of Bloom's, though this is not always linear and should be part of a "bigger picture" that develops when assessing learning within the project.

- the learning outcomes that follow these principles will need to be directly related to the computing aspects of the project, or this approach would simply assess a pupil's ability to learn, rather than to "do, know, understand" Computing
- Other useful approaches along these lines that can be used to inform the assessment a school develops include the Technology Triangles and the Digital Bloom's Taxonomy