



MILES BERRY PRINCIPAL LECTURER

PROJECT WORK: DOES IT WORK?

Miles Berry questions whether GCSE coursework is fit for purpose, and proposes an engaging alternative

I'm a firm believer that computing is a practical subject, having at least as much in common with art, design and technology, and music as it does with mathematics and the (other) sciences. Papert's great insight, that we learn best through making things to show to others, seems to apply to computing as much as any other subject. I believe, although I've not as much evidence as I'd like for this, that practical programming projects are the best way to learn the fundamental principles of computer science.

The problems with the NEA

All that said, I'm far from convinced that the NEA (non-examination assessment, i.e. coursework) components of GCSE computer science are really what any of us would want. There are a number of problems, for example: the tasks are far from authentic or interesting, and allow little scope for creativity. Teachers and pupils find themselves in morally conflicted positions, due to the artificial conditions imposed on the conduct of the assessment. Even the exam boards have doubts over the reliability of this component of the the assessment. Computing is the only English Baccalaureate subject with an NEA, and large numbers of independent schools (who have the choice) choose to enter students for IGCSE computer science, which is assessed through exams only, rather than the GCSE.

Learning from the other sciences

What then can be done? Similar issues arose over assessment of practical work for GCSE science; their solution might work for us too. For the other sciences, there are a list of required practical tasks, which pupils have to undertake, and which the school has to confirm

to the exam board have been completed at a required standard. These don't count for grades, but the concepts and practical techniques covered in these tasks provide content for written questions on the exam papers. For computer science, we could imagine a number of specific programming challenges (e.g. write a program to sort a list of a million 32-bit integers; write a program to compress a text file), that have to be completed, and which might be asked about in the exam. Although many bemoaned the removal of practicals from GCSE science, the result of this shift has been a strengthening of practical, experimental science in schools. Similarly, the independent schools I know that offer IGCSE computing are doing more practical programming, not less.

Show us what you can do!

An alternative would be to go for something much more open-ended. Instead of 'solve this problem subject to these constraints', what about 'show me what you can do'? This does seem to work for A level computer science (and similar qualifications at 18+, like the EPQ and AP CS Principles in the US): students get to choose a worthwhile, challenging project; have scope for independent, creative work; and get something to talk about at interview. There are some great examples out there, more than a few involving Raspberry Pis. I wonder if this sort of open-ended, student-led project could work for GCSE, too? (HW)

Miles Berry is principal lecturer in computing education at the University of Roehampton. He helped draft the English computing curriculum and is a Member of the Raspberry Pi Foundation.