Non-Examined Assessment for GCSE Computer Science

Simon Peyton Jones with Tim Oates

Version 3, 3 December 2018

1 The importance of practical work in GCSE Computer Science

This paper makes the case for a new approach to Non-Examined Assessment (NEA) for GCSE Computer Science.

We take for granted that practical work in computing, particularly programming, is of central importance to the subject. Yet the difficulties surrounding its assessment (Section 3) have sucked the life out of practical work that should be joyful, creative, and insightful.

Our goal is to reinvigorate practical work as a central component of a GCSE in Computer Science.

This paper represents the personal views of the authors, and not of the organisations they work for.

2 The context

It is helpful to understand the roles of various organisations in GCSE examinations.

- The Programmes of Study for computing is set by **Department for Education** (DfE), and specifies what should be taught in computing, over the entire Key Stage 1 to 4 range. It says nothing about how it should be taught, or how it should be assessed.

- The DfE is also responsible for the GCSE subject content for computer science[1] which specifies the **content** of the GCSE Computer Science. It has sections on "knowledge and understanding" and on "skills".

- **Ofqual** is responsible for the "GCSE Subject Level Conditions for Computer Science"[2] which provides guidance on **assessment objectives** for the GCSE in Computer Science.

- The **awarding organisations** (such as AQA, OCR, Edexcel, etc) offer a GCSE in Computer Science that must meet the DfE's subject content, and meet Ofqual's assessment objectives. They each develop a **specification** (which is stable across several years) and **exams** (which obviously change from year to year) that meet that specification. Awarding organisations are sometimes called "exam boards", but we use the former term.

3 Difficulties with directly assessing practical work

In the past, practical work for GCSEs in computing has been internally assessed and externally moderated by awarding organisations. That is:

- The practical work is delivered by centres
- Then marked by teachers and then...
- That mark counts for some proportion (e.g. 20%) of the overall GCSE mark.

This obvious-sounding approach has turned out to have major problems in practice:

[1] https://www.gov.uk/government/publications/gcse-computer-science
[2] https://www.gov.uk/government/publications/gcse-9-to-1-subject-level-conditions-and-requirements-for-computer-science

- **Focus in the wrong place:** Because of the high stakes, the focus of both teachers and students is on achieving the highest mark possible for the candidate to support their written examination. The goal becomes “getting the tick” rather than learning. Once that is done, the task is shelved, and we move on.
- **The consequences of moderation**. Formally-assessed work must be open to moderation by the awarding organisation, and in practice that means (a) it must be in writing, and (b) it must be attributable to an individual student. So, teachers and students must build a portfolio of written evidence – screen shots, test plans, requirements analysis, documentation, flow charts, evaluation studies, and so on – to demonstrate that they have indeed done the work. This has two related consequences
    - Building the portfolio takes a *lot* of time that could be far better spent.
    - The design of the task itself is skewed to meet the need for written evidence of practical work.
    - Practical work is done in a way that is almost diametrically opposite to the processes that are followed in the work place. For example, teamwork is not allowed, nor peer support, nor gathering resources from the internet.

  All of these undermine the validity of the assessment. (I.e. does it assess the skills that we want assessed?)
- **Widespread malpractice:** Due to pressure on attaining results, solutions appear (on the internet, underground or similar) within hours of tasks being published. It is effectively impossible to prevent this happening. There is also a perception that there is only one ‘ideal’ solution that an awarding organisation is seeking.
- **Administration overheads:** To mitigate malpractice, awarding organisations required schools to put in place onerous and time-consuming measures, especially sequestered computer suites and materials.
- **Moral hazard:** Every teacher wants to follow the spirit of the law, but that comes into conflict with their equally legitimate desire to help their students succeed. That leads teachers into moral dilemmas that consume their time and energy.
- **Malpractice procedures:** Awarding organisations are required to play judge and jury on allegations of malpractice. These are hugely demanding on precious teacher time, and emotionally destructive even when teachers are ultimately exonerated. Moreover, the increase in suspected malpractice and associated news reports is undermining the integrity of both the qualification and the teachers delivering it.
- **Consistency of marking.** It is difficult to consistently mark widely varying work; and yet the purpose of rich coursework is to engage with contexts that motivate learners, and hence should be highly varied in their form and content.

In short, the assessment tail is wagging the learning dog.

These difficulties, which have become increasingly acute, led Ofqual to abandon inclusion of NEA as a 20% contribution to the final GCSE grade in mid-stream, during the 2017/8 academic year. The question is: what should replace it?

4 Principles

4.1 Why is practical work important?

- The second of four Aims of the Programme of Study for Computing[3] says that every pupil should be able to "*analyse problems in computational terms, and should have repeated practical experience of writing computer programs in order to solve such problems*"
- Programming makes the abstract concepts of computing both concrete and tangible.
- Programming is highly creative and motivating: students can design and build things that no one has ever built before and share them with their friends, and beyond!
- Programming is an extremely useful life skill.
- "Programming" does not just mean the skills of a professional software developer. Its core concepts and techniques are useful in many other contexts, including website design, scripting of various sorts, data science, visual programming environments, and the natural world (e.g. what "program" is this ant following?).

4.2 What should be the goals for practical work?

Goals for practical work in computer science:

- **Schools and teachers should have incentives to offer practical work in computing**, and students should have incentives to engage with that practical work.
- **Practical work should be primarily for learning**. The overheads of delivery should be as low as possible.
- **Practical work should be done in as realistic a context as possible**, allowing teamwork, peer debate and critique, use of books, and the internet.
- **Practical work should be integral to good pedagogy**, rather than viewed as a vehicle for assessment.
- **Practical work should foster curiosity and exploration**, framed in a structured and safe way to allow the development and consolidation of computer science principles.

4.3 What operational skills do we want students to learn through practical work?

Many people think of "programming" as synonymous as "writing a program from scratch", but practical work in computing can be much richer than that. (e.g. see Mark Guzdial *"What's the best way to teach computer science to beginners"*[4], Sue Sentence *"PRIMM: exploring pedagogical approaches for teaching text-based programming in school"*[5].)

For example, here are some of the skills that a student may acquire through practical work

- Read an existing program and predict its behaviour
- Write test(s) that expose potential bug(s) in a program
- Explain a program's behaviour to someone else
- Find bugs in an existing program: form a hypothesis, figure out how to test that hypothesis, verify the hypothesis; then work out how to fix the bug, or iterate this hypothesis process
- Modify an existing program to do something else
- Write a subroutine for an existing program
- Write an entire program from scratch

---

[3] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study
[4] CACM 58(2) 2015, https://dl.acm.org/citation.cfm?id=2714488
[5] WIPCSE 2017, https://kclpure.kcl.ac.uk/portal/files/79583213/version_pure_primm_1.pdf

- Use abstraction effectively

All of these variants can used to explore competence in

- Variables
- Sequence
- Selection
- Iteration
- Functions and procedures

And many can be explored in a variety of languages and programming environments.

Many of these practical skills are directly informed by knowledge; but this knowledge can be tested in an exam.

It should also be recognised that programming is not necessarily software engineering – which is of wider scope, and involves additional processes such as Analysing Requirements, Design and Testing. A Programmer may be given a set of designs and tests which they will then implement, run, record and pass back to a software engineer for review.

4.4 Programming as a pedagogical vehicle

Programming can drive learning because it is highly motivating (or even addictive) and provides a concrete context in which to explore and experience abstract ideas in action. However, especially for less able learners, programming can be disengaging, challenging, and potentially a block to engaging in computer science.

Pedagogically we should not be developing programmers and assessing their programming "skill" but developing excitement and engagement in computer science concepts through the vehicle of experiencing programming. It should not be enough to say, "John wrote a program" this is a L5 program; rather we should say "John learned X and Y through the medium or writing a program", and be explicit about what X and Y are against the knowledge domain of computer science.

Assessing programming is itself a developing field of research. "Who wrote the best program?" can often lead to a wide variation in views and perspectives, and therefore raises its own validity considerations before we even start to address what we actually set as being of a 'suitable level' of programming skill for a 16-year-old child.

4.5 Real programming?

Despite some claims to the contrary, analysis suggests[6] that we should distinguish "school science" from "real science".

- In real science we see accidents (penicillin), anomalies which turn out to be the main point, years of hard graft (detecting gravitational waves), and super-expensive equipment. Real science is boring, repetitious and time consuming.
- In school science (in contrast) we try to acquire foundational scientific knowledge through carefully constructed situations. We often look to "wow factor" to show science off, which does not necessarily reflect industry and future experience.

---

[6] *The role of practical work in the teaching and learning of science*, Millar R, 2004. Paper prepared for the Committee: High School Science Laboratories: Role and vision, National Academy of Sciences, Washington DC

It's the same with programming. "Real programming" is part of a big life-cycle involving customer interaction; requirements analysis; lots of documentation; software engineering practices such as agile teams, sprints, or scrums; and tools such as bug trackers, build systems, source-code repositories, code analysers etc. It is not appropriate to attempt to teach all of this to a 14-yr-old; there is too much of it, and it is too poorly motivating "in the small".

Instead, students who experience "programming" may well be better served by experiencing practical tasks which construct situations through which students will acquire specific knowledge and skills – and should be explicit about that knowledge and those skills are. The weight of evidence should be the learning journey, rather than the paperwork.

As a case in point – Hayley may create a program that tries to multiply an integer with a real number and store it as an integer. She may spend an hour working out why this failed probably firstly through trial and error, then researching things associated with this, and then probably finally having a nudge from a teacher to locate this error and debug it. To then need to extensively record this through screen shots, prose, redevelopment and then further screenshots slows down that fundamental journey of learning.

And indeed the, the DfE's subject content makes no reference to the software development process, beyond "design, write, test, and refine programs". We just urge that, at GCSE level, the focus should be on "programming in the small" rather than "programming in the large".

4.6 What about students who find exams hard?

One reason that assessed coursework is sometimes welcomed is that it allows students to walk into the exam room with the reassurance of having (say) 20% of the marks under their belt already. Would a move to an exam-only regime discriminate against students who feel this way?

We argue that the reassurance of completed coursework is largely illusory.

All students tend to get good marks for assessed coursework, because they can (and do) invest a lot of effort in it. As a result, the marks for assessed coursework tend to be bunched together at the high end.

But exams are designed to *discriminate*. To take an extreme case, if every student got 50% from their perfectly-completed coursework, all the discrimination would be attributable the exam, even though it theoretically accounts for only 50% of the marks. There would be strong pressure to move grade upwards, to achieve a typical "bell curve" of overall results.

So while marked coursework does give a very strong incentive to complete it, it may not actually help a student who finds exams difficult to get a significantly better grade.

5 Our proposal

Drawing on experience from Science (Section 8), we propose that

- The GCSE is graded on exam marks only. It seems clear that in the current climate – practical work is:
    - Open to abuse – some of which is potentially impossible to stop
    - Not best placed and may have construct-irrelevant variance within it
    - Is significantly onerous in its current format, and is time-intensive in comparison to curriculum time v. tangible 'results'.
- The awarding organisation offers a set of practical tasks, to be done during the course of study, to motivate, illustrate, and bring to life the theoretical aspects of the subject (Section

5.2). Schools, teachers, and schools are given multiple incentives to engage with this practical work (Section 5.1).

- In particular, all or part of a GCSE examination paper (call it the Practical Paper) is explicitly designed so that *students have a much greater chance of success if they have done substantial practical work* (Section 5.3).
- Practical work could additionally be incentivised by an endorsement model of some sort (Section 5.4).

The goal is to **incentivise** practical work, but not to **assess it directly**.

The science GCSE has taken this approach, with very promising initial results, as we review in Section 8. Whilst having a different construct base, it remains the case in computer science that knowledge is an essential component of performance, and therefore the science experience gives very helpful background evidence.

5.1 How can practical work be incentivised?

A worry about our approach is that schools, teachers, students and parents may say "if practical work is not going contribute directly to my mark, I won't do it". This is a substantial concern, and for this very reason the changes in science were very controversial.

Nevertheless, we believe that a combination of incentives will suffice to give schools and teachers the backing they need to offer well-supported opportunities for practical work. They all recognise the educational importance of practical work; all we need do is convince them that devoting time to it will enhance, and not damage, their students' chances of success in the exam.

We envisage a combination of incentives

- **A clear statement of educational principle and intent, from Ofqual,** stating the crucial importance of practical work, and explaining why the absence of direct assessment offers space for much greater flexibility and creativity in its delivery.

- **A clear statement from Ofsted** that, during their regular inspections, they expect to see evidence of well-supported practical work in computing and science. Such is Ofsted's influence that such a statement could have a profound effect.

- **An endorsement requirement** whereby schools must explicitly affirm that their students have been given timetabled and supported opportunities to engage with specific practical work. Each student should be verified as having engaged with (or "completed") the practical work. We say more about possible endorsement models in Section 5.4.

- **A Practical Paper as part of the GCSE**, whose explicitly-stated goal is that candidates should draw on skills developed within the practical work to be successful within it. We believe this is easier to achieve in computing than in science; for example, students can write code fragments in an exam, but they cannot perform a titration. But even the explicit declaration of intent (irrespective of its actual effectiveness) will give schools and teachers a clear reason to offer practical work. We discuss the Practical Paper further in Section 5.3.

- **Situated cognition**. Research shows that children remember principles better if they are demonstrated and exercised in concrete context(s). This work could be summarised so that teachers and schools could point to research evidence supporting their investment in practical work. We cannot overstate the importance of this kind of communication to teachers, so that they understand and support the underpinning rationale for the new assessment and learning model, and can communicate it to others.

- **Smaller focused practical tasks** which do not rely heavily on documented evidence (which was required for external assessment) relieves administrative burden and facilitates more time for repetition and development of skills – which is the aim of practical computing in the first place.

5.2 The nature of practical work in our model

We envisage that each awarding organisation will specify, as part of the materials for their GCSE, a set of practical tasks:

- All students should have well-supported opportunities to carry out these tasks, during their studies leading up to the exam.
- Schools certify that their students have completed these tasks (see Section5.4).
- The learning objectives of each task should be clearly articulated.
- The tasks should be specified in enough detail that a teacher could just use them without modification, with confidence that they meet the learning objectives.
- Teachers should be free to design new or adapt existing tasks, to suit the capabilities and interests of their own learners, provided they meet the learning objectives. After all, the practical work is not marked, and learning is paramount. Flexibility will help engagement from teachers and suit the learners more readily.
- Tasks should not all be of the form “write this program from scratch”; see Section 4.3.
- We do not advocate a “20 hours of practical work” clock, because this encourages a false dichotomy between theory and practice. However, each set task should have in mind a time-frame for completion, and they should share this with teachers e.g. “This activity should not take more than 2 hours of a student’s time”.
- Practical tasks should, where possible, have a “wow factor” rather than be dull, circumscribed exercises (e.g. write a program to add up the numbers between 1 and 100).

Freeing practical work from direct assessment allows much more flexibility and creativity in its delivery. For example:

- Teacher advice is not just allowed, but encouraged.
- Peer advice, where stronger students help weaker students to the benefit of both, can be encouraged.
- The internet is fully available. For example, practical work can use online data sources (say, of potholes in a city) so that students build more meaningful and motivating programs.
- Teamwork/Peer programming is encouraged.
- Failure is OK. Learning, rather than achievement of a working program, is the priority.

5.3 The Practical Paper

We believe that it is feasible to design a written exam in which it would be difficult to do well without having engaged with the practical work specified by the awarding organisation. Indeed computing lends itself more readily to such examination than, say, science. For example, an exam cannot possibly evaluate a student’s ability to perform a titration effectively; but it can ask students to read program fragments, reason about them, and write sections of code.

Repeated practical experience of tackling programming problems builds practical mastery of theoretical concepts – which is the aim of the practical tasks set by the awarding organisations. That mastery allows students to therefore perform more strongly in this paper in comparison to students who have not completed these exercises.

5.3.1 Linking the paper directly to tasks:

An obvious possibility is that the Practical Paper could include questions that refer directly to specific practical tasks that the student should have done; not having done those tasks would disadvantage the student. Although attractive in some ways, we do not advocate this approach.

We would recommend that the Practical Paper should stand by itself, without relying on the specifics of the practical tasks, for the following reasons:

- Relying on practical tasks would force all schools to undertake the specific tasks designed by the awarding organisations, this will limit the ability of schools to define these tasks to suit their students or undertake other tasks with the same learning objectives.
- Questions on set tasks become predictable. The threat to validity of this paper would be immediate. With a growth in the market for "worked solutions" the incentive to analyse this 'pre-release material' to predict what may be asked in the exams, sell it/use it and then to 'teach to task', must be avoided.
- The exam should be free to describe new scenarios/contexts, within which the student is expected to apply and deploy their knowledge, rather than relying on only those from the set tasks.

5.3.2 A separate Practical Paper?

Rather than having a distinguished "practical paper", it would be possible to integrate the assessment of practical skills throughout all the examination papers. Different awarding organisations might make different choices here: specifying the number and organisation of papers is beyond what Ofqual should do.

However, we think that it is helpful to have the additional clarity of having a paper whose declared purpose is (in part at least) to test operational skills and knowledge. We also would recommend that the time-space that candidates have to engage with the paper is recognised as potentially needing to be more than would otherwise be given to a more 'traditional' exam paper.

5.4 The endorsement model

We propose that: each school certifies, for each student S, whether S has meaningfully engaged with N of the tasks specified by the awarding organisation for that GCSE. Note that:

- It is not required that students "get the right answer" in every task. Rather, "meaningfully engaged with the task" means that the student made a serious effort to understand and execute the task.
- The certification bar is deliberately low: it asserts only that the student has made a decent attempt, and not how good that attempt is. It's up to the written exam to measure level of attainment.
- It is a requirement of the teacher to ensure that each student has meaningfully engaged with the practical work, in a way that is appropriate to their specific learning needs. The school must submit and annual declaration of compliance in support of their certification.

- The awarding organisation could perform some sampling to check that schools are submitting truthful declarations; but well short of formal monitoring visits to a substantial proportion of centres. The reliability of the qualification is based on the exam.

Overall, our intent is that the endorsement should provide an incentive for schools to offer well-supported opportunities for practical work, and for students to engage with that work; but that the endorsement process should have very low overheads. Unscrupulous schools could indeed lie, but doing so would not improve their students' outcomes, so the incentives to do so would be absent.

6 Variations on the theme

In this section we explore two additional possibilities. The section is more speculative in nature and does not form part of our core recommendations.

6.1 Additional recognition for outstanding practical work

Teachers and students may regret the lack of any form of recognition (in the exam result) of outstanding programming effort or ability. There are various ways in which this might be addressed

- One way to address this might be for the awarding organisations to offer some sort of additional endorsement for candidates who create a significant program to a brief and submit it for evaluation. The tasks could be similar in nature to the current ones offered for the current programming project and set by the awarding organisations. The emphasis would be on creating a significant, functional working program that fulfils the brief and demonstrates evidence of completion/functionality.

  We would encourage such awarding organisations to offer a simple Pass/Fail endorsement, rather than have a complex grading structure. There would be no impact on the grading of the GCSE itself – but could be recognised by a Practical Endorsement 'Pass' on their exam transcript, or by a separate achievement award.

- The Extended Project Qualification is well established at age 16-19. It would be worth exploring whether something similar could be done at age 16, dedicated to computer science.

- Awarding organisations could offer infrastructure for a structured online portfolio that, as in art, design, and engineering, students can carry with them.

- Awarding organisations could run a national competition.

6.2 On-screen exams

Our principal recommendation is that the formal assessment should be under exam conditions. An entirely viable possibility is to conduct such an exam on-screen, allowing students to write and execute programs, with immediate feedback about trivial errors and the familiar support of their IDE. The current Edexcel IGCSE in computer science Paper 2 is conducted in this way.

Opinions differ about whether on-screen examination would be a good thing or not, and (although it seems plausible) we lack research evidence to favour one path or the other. The idea is certainly not central to our proposal.

An on-screen exam would impose substantial logistical challenges, although if evidence emerged to show that it was better, those challenges could probably be overcome. We would also question whether we are testing programming competence or assessing knowledge domain through programming – again coming back to construct, and potential for construct-irrelevant variance.

Further research into on-screen testing and its pedagogy for assessing programming skill should be undertaken.

7 Programming languages

An exam involving programming must use *some* notation to communicate program fragments. Is worth making a careful distinction between three distinct purposes:

- The language in which examiners pose questions to students (Section 7.1).
- The language in which students write answers to examiners (Section 7.2).
- The language in which practical tasks are carried out, during the course, and not formally assessed (Section 7.3).

In this context people often speak of "pseudocode". It is helpful to distinguish two quite different forms of communication:

A. **Pseudocode**. An informal notation used for sketching a program and explaining it to others. It may contain suggestive (but manifestly imprecise) English-language statements such as "get the data from the database" or "loop over all items in the list". For this purpose, any notation will do; it is by-design imprecise, and the usual expectation is that the author is present to discuss it.

B. **Exam Reference Language**. A formal notation that examiners can use to express a program, or program fragment in an exam question. There is a tendency to call this "pseudocode" too, but doing so causes fatal confusion. Unlike pseudocode, *the language in which examiners pose questions must be fully precise.* There is no room for informality or imprecision, since it is crucial that students are well prepared to understand it. We strongly urge that the term "pseudocode" should not be used for this form of communication.

7.1 The language in which exam questions are expressed

The language used for program fragments written by examiners in an exam question could be:

- **A completely new Exam Reference Language designed by the awarding organisation**. A possible advantage of this approach is that the language can be tailored to the purpose, whereas (even a subset of) existing languages are optimised for programming at scale. And awarding organisations are understandably reluctant to privilege any one existing language over another.

- **A carefully-specified subset of an existing language**. A big advantage of this is that students do not need to learn another new language; instead they can focus on an existing language, with its supporting ecosystem of IDEs, books, tutorials, and so on.

  Rather than choosing a single language, the awarding organisation could specify a small **set of languages** (or rather, well-specified subsets thereof). Each exam question that presented code be offered in alternative forms, one for each language. This would take a bit more work, but not much compared to the effort of producing the question in the first place; it would give more flexibility to teachers; and it would avoid the awarding organisation "blessing" one particular language.

It should be up to the awarding organisation what they choose. But we do urge:

- That awarding organisations give serious attention to the possibility of using a well-specified subset of an existing language or languages.

- That any awarding organisations that choose to design an Exam Reference Language should collaborate on its specification, to make it easier for schools to move between qualifications.
- That any such Exam Reference Language should never be called “pseudocode”.

7.2 The language in which students write answers

A related question is this: *in what language may students write answers*?

- In general, questions should test the students’ understanding of *semantics*, rather than their recall of *syntax*. For this reason, most awarding organisations specify a set of languages that they may use; and they ask examiners not to be picky about syntax (unless that is the point of the question).

- For some questions – such as those asking students to design a program – it may be more appropriate to invite students to respond using a flowchart, or an informal pseudocode, provide the student’s intent is clear to a reasonable examiner.

7.3 Programming languages for the practical tasks themselves

Under our proposals, awarding organisation would provide practical tasks to be done during the course, and (although they would not contribute to the grade) there would be strong incentives for students to undertake these practical tasks.

The question arises: what programming languages should be used for these tasks?

7.3.1 Textual and visual languages

The Programme of Study[7] says that pupils should be taught to *“use two or more programming languages, at least one of which is textual, to solve a variety of computational problems*”. The DfE GCSE subject content for computer science[8] requires that students should develop skills including “*design, write, test and refine programs, using one or more high-level programming language with a textual program definition, either to a specification or to solve a problem*”.

Accordingly, the Ofqual “GCSE Subject Level Conditions for Computer Science”[9] require awarding organisations to “*set out what programming language the learners are permitted to use for the purposes of the programming project*”, and specify that “*each such programming language is a high-level programming language that has a textual program definition*”.

Under our proposal, pupils would be forced to become proficient in a textual language in order to understand and answer the (necessarily textual) Practical Exam.

But for the practical tasks themselves there should be no formal requirement to use a textual language. Such a requirement would be hard to enforce, since the practical work is not assessed; and some students may be best served by using visual languages for the most part. If the aim is to develop understanding of programming principles, then creating a loop in Scratch requires just as much pedagogical understanding of the concept of loops, as creating one in C#.

[7] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study
[8] https://www.gov.uk/government/publications/gcse-computer-science
[9] https://www.gov.uk/government/publications/gcse-9-to-1-subject-level-conditions-and-requirements-for-computer-science

7.3.2 Graphical user interfaces

Just because the *programming language* is textual does not mean that the *program* must have textual input and output; with a suitable library, practical exercises with a textual programming language can easily involve rich graphical applications.

8 The experience of science

We may learn a lot from looking at recent experience in science, which has pioneered the approach we propose for computing.

8.1 What happened in science

The science model was a response to a set of serious problems in using marks from practical work to contribute to the overall grade. The problems were:

- **Endemic maladministration/malpractice**. Now widely acknowledged (see TES 23 09 16 p12) teachers were under great pressure to elevate marks from the practical and/or engage in maladministration. Attributed to the effect of high stakes accountability, the maladministration shows in the data as an endemic problem.
- **The failure of 'controlled assessment'**. The first attempt to curb maladministration/malpractice and distorted marks was a QCA move to 'controlled assessment' - tightening the tasks and activities and the conditions of administration. This failed to achieve the objective and possessed a collateral impact of making the practical assessments extremely boring and possessing a high workload for teachers.
- **Professional contradictions and breakdown of trust**. When systemic incentives drivers push teachers into maladministration/malpractice, something is going seriously wrong in education. With coursework and practicals contributing marks to the grade, teachers were under pressure to both attain increasing grades year after year, and also act as the objective agent of an awarding organisation in making the assessments. This was an unfair and untenable professional contradiction, which erodes trust between State, teachers and awarding organisations. Additionally, Reiss and Abrahams' research showed an escalation of teacher and pupil cynicism towards practical assessments.

This severely dysfunctional state of affairs needed fundamental action. Practical work needed protection, and needed to be re-constituted. Teachers needed to be trusted professionals running high quality programmes which included a significant amount of well-structured practical work. The educational outcomes of practical work need to contribute to the grade.

During the development process, researchers examined all the evidence, consulted with higher education science departments, and used fundamental work on the importance of practical work to make far more clear the educational purpose of practical work, which in most instances was acquisition of knowledge of principle, experimental design, recognition of measurement accuracy, understanding observation and recording, and conceptual understanding. The majority of these key constructs could be assessed in examinations, while the best method of attaining is most obviously to provide rich practical work, not narrow drilling.

This resulted in two tightly related models for A level and GCSE science which:

- had a segment of the examination which included questions indirectly assessing the outcomes of practical work. This is a fundamental aspect of the model: the theory- and evidence-driven principle that one core objective of practical work is the acquisition of knowledge. This is not an impoverished ideas of 'facts' but 'knowledge essential to skilled activity'. Knowledge is intrinsic to performance, not separate from and unrelated to skilled performance. And this in addition to the purpose of practicals as demonstration of laws, relationships of cause and effect, of the operation of underlying processes, etc. Such complex and rich knowledge can be assessed by examination and has successfully been assessed in this manner for centuries.

- stipulated that schools should complete a number of practical activities from a stated list, and emphasised the link between these and the constructs assessed in the examination. The school would sign a statement to confirm that students have had the opportunity to do a range of practical work that cover stipulated apparatus and techniques.
- for A level, but not GCSE because of volumes, included monitoring visits by the awarding organisations to ensure that high quality practical work was being undertaken.

The model was constructed carefully and as a coherent whole; the impact of each part being considered carefully, the interactions and incentives modelled, and every element being research-driven. There were no ad-hoc or panicked elements. Immediately controversial, the model was understood and adopted as national policy by Ofqual.

8.2 How the new scheme worked in practice

Immediately controversial, the model adopted by Ofqual resulted in public outcry. Ofqual officers, particularly the chief executive Glenys Stacey and the chair Amanda Spielman, displayed a resolute commitment to the model, rightly convinced of the strength of the evidence and theory underpinning it. Learned societies and science educators almost universally predicted that this would cause a calamitous collapse in practical work. There was anxious public discussion and questions asked in Select Committee.

But early monitoring visits in 2016 to OCR pilot centres showed universal support from teachers and pupils; '...we can now learn from practicals, instead of treating them as a stressful way of getting marks...', '...I can learn from mistakes, now...'. The latest overviews by Ofqual and OCR shows that, far from the collapse predicted, the amount of practical work has been sustained, despite severe pressure from other issues such as a shortage of specialist teachers, technicians and funding.

The move to the science model can be regarded as a genuine success, resolving chronic, severe difficulties and erosion of trust, and supporting sound educational practice. For computer science there can be the same well-structured approach to development of valid and dependable assessment of the learning outcomes from practical activity.

The Ofqual blog post Assessing science practical skills at A level, May 2018, summarises early feedback, and links to a variety of useful papers including

- The impact of qualification reform on A level science practical work: Paper 1: Teacher perspectives after one year. Ofqual 2017.
- Paper 2: Pre- and post-reform evaluation of science practical skills. Ofqual May 2018
- Paper 3: Valid discrimination in the assessment of practical performance, Ofqual May 2018

8.3 The purpose of practical work in science

In science, there has been an acknowledged lack of clarity regarding the purpose of practical work. This is a remarkable state of affairs, since science is such a long-established enterprise in education. Scrutiny of justifications suggests the importance of

- acquisition of skills of observation,
- recording data,
- handling materials and equipment,
- working to specification, and

- working with others.

But acquisition of conceptual understanding and development of knowledge also features heavily:

- understanding cause,
- understanding the importance of accuracy,
- understanding physical relations and interactions, and so on.

Some of these show the link between skills and knowledge - for example, handling toxic materials or managing the toxic products of reactions requires complex understanding of the behaviour of the materials (eg chlorine gas) and the degree of harm which can result from different types of exposure.

In addition, there is a body of theory which points to the importance of 'grounded theory' and 'situated cognition'. Specifically, *the experience of practical activity stimulates more secure and more elaborated understanding of a specific concept* - for example, oxidation.

More background on the importance of practical work can be found in these papers

- *The assessment of practical work in school science*, Abrahams I, Reiss M and Sharpe M, Studies in Science Education v49 n2 p209-251, 2013.
- *The role of practical work in the teaching and learning of science*, Millar R, 2004. Paper prepared for the Committee: High School Science Laboratories: Role and vision, National Academy of Sciences, Washington DC
- Good Practical Science, Sir John Holman, Gatsby Foundation, Sept 2017
- Assessing experimental science in 11-18 education. A report of a Royal Society/Gatsby Foundation/Wellcome Trust conference, Oct 2016.

From this it can be seen that a variety of functions can be allocated to practical work in science, not all of them requiring direct assessment during the practical work itself (since the outcome is an enchantment of the pupil's knowledge of conceptual understanding). Moreover, some of the skills may be 'threshold' skills - they need to be proved to be absent or present, rather than 'graded'.

8.4 Assessing practical work in science

Until the revision of science GCSE and A Level qualifications in 2015, little of this evidence on the variety of functions of practical work was taken into account in the development of the assessment model for the qualifications. The new models introduced clarity into the purposes of practical work as well as ensuring that desirable outcomes were distributed correctly across the different requirements of the qualification - hence the emphasis on practical work being undertaken (giving experience of the handling of materials and equipment, observation etc) and assessment in the examination of the conceptual understanding and knowledge which is both required in practical and is either consolidated through, or is the product of, practical work.

This model is both sophisticated in its rationale and grounded in evidence. By contrast, the commonly-stated "practical work gives pupils direct experience of real science" is breathtakingly crude. It neither separates carefully the desired outcomes, nor does it represent the reality of contemporary science.

Unlike real science, practical work in science education has to take place over a short period (usual maximum 80 minutes, although some experiments such as growing plants can occur over a protracted period). Practical work is also amateur in character: unlike professional scientists pupils do not necessarily know what they are looking for but are using the practical work to understand what they should be looking for. It seldom is ‘real science’, and the view that it ***is*** misrepresents genuine scientific enterprise as well as distorting the educational purpose of practical work in schools.

We acknowledge that there are emerging important opportunities for pupils to contribute to real science - eg mass contribution of analysis of astronomical data, participation in wildlife recording. But these are highly distinctive, and again there needs to be clarity in the pupil outcomes which are expected from the activity – exactly which elements of knowledge, skill, conceptual understanding are the intended outcomes of the activity and can they be dependably assessed?

8.5 Lessons for computing

Can the A level and GCSE models for Science be used in their exact form for the Computer Science GCSE? Yes, but with some provisos. There needs to be attention to the purpose of practical work in the educational processes in Computing Science.

Professional activities in the workplace are in the most part practical or require understanding of the practical activity of computing science. This gives practical work a clear place in the curriculum (the learning programme) but does not necessarily mean that there is a place for *assessing* practical work.

- As with science, the pressures from asking teachers to assess with precision continue to be overshadowed by the pressures from accountability and other instructions to ‘improve grades’.
- As with science, the educational purpose of practical work needs to be established and stated with precision.
- As with science, practical work (rather than its assessment) can be stimulated by requirements that certain practical work, and a certain amount of practical work, needs to be completed as a requirement of the programme, even it does not contribute to grading.

Many of the outcomes of practical work are likely to be cognitive in character, including issues relating to the reasons for doing certain things, for taking specific courses of action, and for avoiding risk and improving effectiveness. These can be assessed in a formal written/on-screen examination.

One further issue which renders practical work leading to grade problematic in computing science is the issue of system, language and context. Variations in these do limit the extent to which a person can work effectively in one context but not in another. Including restricted applications, or specific setting and systems can adversely affect the motivation and engagement of pupils, the generalisability of the assessment, and may favour one group of pupils over another. This leads to poor measurement by the assessment, and severe issue of validity. Again, this pushes in the direction of programmes rich in practical work but with assessment of cognitive outcomes being focussed on, and limited to, the formal examination.