

CONSULTATION

# Future assessment arrangements for GCSE (9 to 1) computer science

**ofqual**

# Contents

|           |   |           |
|-----------|---|-----------|
| <b>1.</b> | <b>About this consultation</b>  | <b>3</b>  |
|           | <i>Duration</i> .....   | 3         |
|           | <i>Respond</i> .....  | 3         |
| <b>2.</b> | <b>Introduction</b>   | <b>3</b>  |
| <b>3.</b> | <b>Summary of our proposals</b>   | <b>4</b>  |
| <b>4.</b> | <b>Background</b>   | <b>7</b>  |
| 4.1       | <i>The issues</i> .....   | 7         |
| 4.2       | <i>Our response</i> .....   | 7         |
| 4.3       | <i>Thinking longer-term</i> .....                                       | 8         |
| 4.4       | <i>Our findings</i> .....   | 8         |
| <b>5.</b> | <b>Options Considered</b>   | <b>9</b>  |
| 5.1       | <i>Assess programming skills through NEA</i> .....                      | 9         |
| 5.2       | <i>Separately report an endorsed grade for programming skills</i> ..... | 11        |
| 5.3       | <i>Assess programming skills by examination</i> .....                   | 14        |
| <b>6.</b> | <b>Our View</b>   | <b>15</b> |
| <b>7.</b> | <b>Impact of Our Proposals</b>  | <b>17</b> |
| 7.1       | <i>Equality Impact Assessment</i> .....                                 | 17        |
| 7.2       | <i>Regulatory Impact Assessment</i> .....                               | 18        |
| <b>8.</b> | <b>Consultation Questions</b>   | <b>20</b> |
|           | <b>Annex A: Proposed changes to our rules for GCSE computer science</b> | <b>22</b> |
|           | <i>I. Remove requirements for programming project</i> .....             | 22        |
|           | <i>II. Revised assessment objective weightings</i> .....                | 23        |
|           | <i>III. New rules for assessing programming skills</i> .....            | 23        |
|           | <i>IV. Our proposed rules</i> .....                                     | 23        |
|           | <b>Annex B: Your data</b>   | <b>30</b> |

# 1. About this consultation

This consultation is about how students taking GCSE computer science should be assessed. The consultation follows interim changes we made to the qualification's assessment arrangements earlier in the year. Our proposals will be of particular interest to computer science teachers and their students, although we welcome responses from anyone who has an interest in the qualification.

## Duration

This consultation will be open for 4 weeks starting on **Monday 5 November** and ending at **4pm on Monday 3 December**. We will announce our decision in **January 2019**.

## Respond

Please respond to this consultation using one of the following methods:

- complete the online response available on [this webpage](#).
- email your response to [consultations@ofqual.gov.uk](mailto:consultations@ofqual.gov.uk) - please include the consultation title in the subject line of the email and make clear who you are and in what capacity you are responding.

For information on how we will use and manage your data, please see Annex B.

# 2. Introduction

GCSE computer science students must develop practical programming skills. The content for the subject<sup>1</sup> requires students to develop their ability to design, write, test and refine computer programs using a high level textual programming language. These abilities should, therefore, be assessed.

We originally intended that students' grades for the qualification should be determined by their performance in a programming project as well as in their exams. We intended the project (a form of non-exam assessment (NEA)) should provide 20% of the total marks and the exams 80%.

In January 2018, in response to concerns about malpractice in the way some students were approaching the project, and following consultation, we announced that exam boards would [assess students taking the qualification in 2018 and 2019 by exam alone](#). Students would also complete a project set by their exam board, but their performance in the project would not contribute directly to their grade. We later [extended this arrangement](#) to students taking their exams in 2020 too.

When we announced these short-term changes we made it clear in our [decisions document](#) that we would consult on the longer-term assessment arrangements for the qualification.

This consultation does not concern the subject content for GCSE computer science, which is set by the government. Programming will remain a key feature of the

---

<sup>1</sup>[Computer science: GCSE subject content, January 2015](#)

qualification. Rather, in considering the options for the longer term, our priority has been to ensure the assessments validly assess all of the subject content in a way which contributes towards students' final grades.

We have been mindful of the potential burden on teachers of different assessment arrangements. In considering the timetable for introducing any new arrangements, we have also recognised the need to allow teachers and exam boards adequate time to prepare.

We have waited until now to consult on the longer-term options so we could take account of any issues with the way the short-term arrangements worked this year. Nothing arose that gave us cause for concern. The association Computing At School (CAS) found that students who had properly engaged with the project should have been better prepared for their exams than those who had not, [as the exams assessed students' understanding of programming](#). However, some teachers have told us they would like more freedom than the interim arrangements allow, in order to determine how to manage their teaching time and how they develop their students' programming skills.

We have discussed with teachers and subject associations the potential ways by which students' programming skills could be assessed in the longer term.

We have concluded that it is not possible to use non-exam assessment in this qualification to assess programming skills in a way that is manageable, reliable and fair. We have also concluded that exam boards could assess programming skills in different and potentially innovative ways under exam conditions (whereby all students take assessments set and marked by exam boards at the same time, under supervision). Moreover we think this approach is likely to lead to the most effective development of programming skills, as it allows teachers to adopt the approach to developing those skills they feel is best for their students.

To allow exam boards to develop their approach (potentially utilising on-screen or other innovative forms of assessment) and teachers time to prepare to teach students who will take any new assessments, we propose that the new arrangements should first take effect for students who will begin their studies in September 2020 and who will take their exams in summer 2022. The interim arrangements would continue until then.

**We are seeking responses to this consultation by 4pm on 3 December 2018.**

### 3. Summary of our proposals

For students being assessed for the first time in 2022 (i.e. whose teaching will start in 2020), we propose that:

- All assessments are conducted under exam conditions, including the assessment of programming skills. Our definition of exam assessment<sup>2</sup> will allow for assessments to be on-screen, including online, if that is an exam board's preferred approach.

---

<sup>2</sup> Ofqual's definition of assessment via examination is "exams that are taken by all students at once, under formal supervision, and are set and marked by exam boards".

- At least one of the exams will require students to complete the following steps to a set task or to solve a problem<sup>3</sup>. Each of these steps could be undertaken as separate activities to separate tasks or to solve individual problems, or addressed in combination:
  - i. Design a program – using one or more high-level programming language with a textual program definition.
  - ii. Write a program – using one or more high-level programming language with a textual program definition.
  - iii. Test a program – using one or more high-level programming language with a textual definition.
  - iv. Refine a program – using one or more high-level programming language with a textual program definition.
- The current short-term arrangements for assessing programming skills remain in place for students taking their exams in 2021. These arrangements require students to have 20 hours set aside in their timetable to complete a programming task. Exam boards assess students' understanding of programming in the exams.

We do not propose to set any further requirements on the form of the exams, beyond those already contained in the subject-level conditions<sup>4</sup>. This would enable exam boards to decide whether in their exams, they:

- set a single task-based assessment.
- set an examination question or series of questions on one or more tasks/problems that together cover the requirements.
- deliver the assessments on paper or on-screen – and if on-screen, whether online.
- use pre-release materials to give students advance notice of the context to a task/problem that will be set in the examination and/or allow students before the exams to work on a program/code about which they are then examined.
- adopt another approach to assessment which meets our rules.

We also propose to reinstate the original assessment objective weightings for GCSE computer science (i.e. those specified when the reformed qualifications were first designed). This will reflect the intended focus on the assessment of programming skills within the qualification. We changed the weighting of the assessment objectives when we removed the non-exam assessment from the final grade, increasing the weighting of AO1 and AO2 and reducing the weighting of AO3.

By returning to the original assessment objective weightings we will re-establish the emphasis in the examinations on students' ability to analyse problems in computational terms. It will be for teachers to decide how best to develop their students' understanding of programming and their programming skills, in the same way teachers can decide how best to develop the key subject requirements in other GCSEs.

---

<sup>3</sup> Section 5 of the published content produced by the DfE requires students to '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'.

<sup>4</sup> Subject-level conditions are the detailed rules we expect exam boards to meet in the design and delivery of qualifications in this subject.

The original, interim and proposed assessment objective weightings are summarised below:

| AO  | Assessment Objective  | Original Weighting | Interim Weighting | Proposed Weighting |
|-----|---|--------------------|-------------------|--------------------|
| AO1 | Demonstrate understanding of the key concepts and principles of computer science  | 30%                | 35 – 40%          | 30%                |
| AO2 | Apply knowledge and understanding of key concepts and principles of computer science  | 40%                | 45 – 50%          | 40%                |
| AO3 | Analyse problems in computational terms: <ul style="list-style-type: none"> <li>• to make reasoned judgements.</li> <li>• to design, program, evaluate and refine solutions.</li> </ul> | 30%                | 15 – 20%          | 30%                |

Returning to the previous assessment objective weightings would also remove the need for a range of weightings. This was necessary because each exam board had taken a slightly different approach to the allocation of the marks for each assessment objective in their exam papers, and thus the removal of the NEA had different impacts on each exam board’s specification.

We have considered whether to introduce a requirement for a written statement from the head of centre. This would confirm that students had been given the opportunity to design, write, test and refine programs, using one or more high-level programming language with a textual program definition as part of their GCSE course. Such an arrangement would reflect the requirements we have put in place for GCSEs in the sciences, geography, geology and citizenship. As students will have to receive programming instruction to be prepared for the examinations, this might be an unnecessary additional burden. On the other hand, we have heard from some stakeholders that a requirement for such a statement would provide additional assurance that appropriate attention is given to programming skills during the course. We are seeking views on the merits of such a statement through this consultation.

To implement our proposals, we will need to change our rules for GCSE computer science. These proposed changes, which are set out in detail in Annex A, would:

- revise the assessment objective weightings in line with our proposals above.
- remove the existing requirement for schools and colleges to set aside 20 hours for students to undertake a programming project.
- subject to our decision following this consultation, potentially remove the related requirement for exam boards to collect formal statements about these projects from schools and colleges.
- implement our proposed approach to assessing programming skills.

## 4. Background

### 4.1 The issues

We originally intended that students taking 9 to 1 GCSE computer science should complete a task, set by their exam board, to test their programming skills and their knowledge of the development cycle. The tasks were designed to be completed in 20 hours under tightly controlled conditions and to contribute 20% of the marks for the overall qualification grade. In autumn 2017 we saw evidence that some of the tasks had been posted to online forums and collaborative programming sites.

Detailed solutions to the tasks quickly became available online, with some viewed thousands of times. We were concerned about the apparent level of malpractice relating to the NEA, the potential impact on fairness for all students and about the validity of the qualification.

We were also aware of teachers' concerns about the restrictions which the exam boards had put in place around the completion of the NEA. The restrictions were intended to make sure the assessment was valid and fair for all, but they created an approach to teaching and learning which did not align well with real-world approaches to programming. Professional programmers often work collaboratively, solving problems by seeking and building on existing programs, developing solutions with input from others and using the same code repositories that were facilitating student malpractice.

We also heard concerns that the exam boards' rules were placing unreasonable pressures on teachers and having a direct impact on their ability to prepare students for the qualification. Some of those seeking solutions to the programming tasks online appeared to be teachers, concerned about how best they might prepare students for the NEA. As the exam boards' rules prohibited discussion of the task outside of the classroom, this constituted malpractice.

### 4.2 Our response

In November 2017, we launched a consultation on the potential short-term options for assessing programming in 2018 and 2019, to address these issues. Following the consultation, we announced that students would still be required to complete a programming project under controlled conditions in 20 timetabled hours, but that it would no longer contribute to the final 9 to 1 grade. Teachers would not have to mark their students' programming projects, although their feedback would help prepare students for the examination. We also suggested to the exam boards that they could review their rules around how the tasks were conducted, to make them more manageable for teachers and a more authentic and formative experience for students.

We changed the assessment objective weightings to reflect the removal of the NEA from the final qualification grade. We allocated a greater proportion of the marks to AO1 ('Demonstrate knowledge and understanding of the key concepts and principles of computer science') and AO2 ('Apply knowledge and understanding of key concepts and principles of computer science') than to AO3 ('Analyse problems in computational terms to make reasonable judgements, and to design, program,

evaluate and refine solutions'), which the exam boards had opted to assess largely through the NEA task.

The 2018 exam papers had already been written when the malpractice on the NEA came to light, and those for 2019 were well underway. We did not want to require the papers to be changed, because this would have risked errors in new papers created at speed and unsettled teachers' and students' expectations about the exam format. The revised weightings of the assessment objectives allowed the exam boards to use their planned exam papers. We allowed ranges in the weighting for each assessment objective, as the exam boards had not all taken the same approach to the allocation of the assessment objectives across their assessment components.

### 4.3 Thinking longer-term

While we made it clear that we would limit the extent of the immediate changes, we also explained we would be exploring a broader range of options for the assessment of programming skills in the longer-term.

Responses to our consultation on the interim changes included a number of suggestions about how programming skills might be assessed, some of which would take some time to develop, test and implement.

We extended the interim arrangements to apply to students taking their assessments in 2020 to allow us to gather input from as broad a range of stakeholders as possible on potential longer-term assessment approaches.

### 4.4 Our findings

We have met with teachers and interested subject associations, and have spoken to some consultation respondents who had strong views about the way in which programming skills might be better assessed in the future. We also engaged with the exam boards, academics, and industry representatives.

We are satisfied from our conversations with teachers and other stakeholders that the short-term arrangements currently in place to replace the NEA are providing students with a meaningful experience of programming as part of a broader computer science qualification. The analysis<sup>5</sup> carried out by Computing At School (CAS) of the summer 2018 examination papers across all the exam boards supports this view. CAS identified between 24% and 42% of examination marks were awarded for knowledge, skills and understanding which should have been developed by the various programming tasks set by the exam boards. These tasks provided a valuable opportunity for students to develop their understanding of, for example, code constructs and comprehension, debugging, and the writing and modularisation of algorithms, which will have contributed to their exam performance.

We propose that the interim approach should remain in place for an additional year, to allow the exam boards time to develop (and where appropriate test) their assessments. Despite our relatively late decision to exclude NEA from the final qualification grade, exam boards did not encounter problems at awarding in summer 2018 and feedback from schools indicates students achieved the expected grades. We want to ensure that teachers will have time to consider the approaches to

---

<sup>5</sup> [Non Examined Assessment – a positive future](#)



assessment exam boards will use from September 2020 onwards, so they can identify the approach that will best suit their students.

Our discussions with stakeholders illustrated the number of ways by which the required programming skills could be assessed – but no obvious way of using NEA which will successfully mitigate the risk of malpractice. A number of the approaches initially suggested to us by teachers involved NEA in some form – including a return to the 20-hour programming task with further rules and restrictions on how this would be completed. It is clear that to secure greater reliability of NEA would impose additional burden on centres and teachers – and yet the risk of malpractice would remain.

Previous experiences within this qualification have shown that additional rules do not necessarily enhance the quality of assessments or the experience of students and in many cases they do not have the intended effect. Our original decision to include NEA within GCSE computer science was a finely balanced one. Despite the exam boards putting in place safeguards, including a shortened assessment window, monitoring visits, and statistical and online monitoring, it quickly became apparent that these measures would not prevent malpractice, leading to our decision to remove the programming task from the final qualification grade.

## 5. Options Considered

Based on the responses to our previous consultation and our discussions with stakeholders, we considered three broad approaches for assessing programming skills in the longer-term:

- assess programming skills through NEA.
- separately report an endorsed grade for programming skills, alongside the 9 to 1 grade.
- our preferred approach: assess programming skills by examination, which could be on-screen (including online) or on paper.

We believe assessing programming skills by examination offers the most robust way of ensuring the required programming content is assessed in full and included in the 9 to 1 grade. It will protect the validity of the qualification, allow exam boards to take an innovative approach to assessing these skills, and enable teachers to decide how best to develop their students' skills.

Within each of these approaches we considered a number of options in detail. We set out our analysis of the different options below.

### 5.1 Assess programming skills through NEA

#### 5.1.1 Reinstatement of the NEA as part of the final qualification grade, with additional rules (Option 1)

Some teachers argued for the reintroduction of the NEA in its previous form with additional exam board rules and monitoring to detect malpractice and sanction those who committed malpractice. They suggested additional safeguards including greater monitoring and a formal, oral examination by teachers (similar to a *viva voce*) to ensure that students' work was entirely their own. It was argued that if these rules

were enforced in a significantly robust manner the risks of malpractice would be mitigated and the NEA could contribute 20% to the total qualification marks, as we originally intended.

However, we have heard limited support for this approach, primarily because of the burden on teachers both of managing the 20 hours for the programming task and of marking their students' work, particularly if this included an oral assessment.<sup>6</sup> There are also risks to requiring teachers to conduct these forms of assessments to a common mark scheme.

It is not obvious what other rules could be put in place to address the malpractice previously seen in relation to the NEA in this subject. For example, where in 2017 solutions to tasks were shared online, these were not always quickly identified. When they were, it was not always possible for the exam boards to have them removed from third-party websites. We do not believe that any further safeguards will prove effective.

Even if exam boards were able to detect all cases of malpractice, if the scale was as significant as it was in the legacy computing GCSE and as we saw at the start of the 2017/18 academic year, this could prove a significant drain on their resources. If the exam boards were still investigating large volumes of suspected malpractice at the time of awarding, they might have to delay issuing results. This could undermine confidence in the qualification, would be upsetting for students and potentially delay their progression.

### **5.1.2 Have a shorter assessment period with the use of pre-release material (Option 2)**

Some stakeholders suggested the use of pre-release material could enable students to prepare for an extended assessment in which they completed a task under supervision. The NEA in GCSE art uses this approach. The amount of time needed to assess the task could be reduced from 20 hours to something in the region of five hours (depending on the task). It could therefore be completed in one day. It would be reasonably easy for exam boards to implement.

Others did not feel that this would be viable if the assessment were to be taken in one session and on-screen; there would be resource implications for many centres. Some schools and colleges enter large numbers of students who would all require access to computers at the same time. This would place significant demands on schools' ICT and networks and prevent other students using the facilities. Centres would also have to manage any health and safety issues arising from continuous use of a computer over a long time. If students did not all have to take the assessment at the same time this would create some of the opportunities for malpractice previously seen as students could share the task online with others yet to complete it.

---

<sup>6</sup> The evidence of the conduct and marking of oral examinations would be likely to require audio and/or audio visual recordings.

### **5.1.3 Assess students' programming skills through fixed duration residential workshops conducted by independent examination teams (Option 3)**

Some stakeholders told us about the computer programming industry's use of residential workshops to assess programming skills.

However, we do not consider this to be a viable option for GCSE students. It would be overly burdensome to centres and costly. Such workshops would disrupt the school timetable and require teacher supervision. Such workshops might not be suitable or accessible for all students, potentially reducing uptake of the subject.

## **5.2 Separately report an endorsed grade for programming skills**

### **5.2.1 Reinstate the NEA with a separately reported grade (Option 4)**

We considered an approach similar to those used for the assessment of practical skills in A level science qualifications and the reporting of spoken language proficiency in GCSE English language. In this option, students' programming skills would be assessed in a NEA project and the outcome reported alongside their 9 to 1 grade. The 9 to 1 grade would be determined by the students' exam performance alone. Programming skills might be reported using pass/merit/distinction or as pass/fail.

We are not convinced that it is feasible to demarcate programming skills from the rest of the assessed content in computer science in this way, because there is a significant overlap with content already assessed in the exam. A student's programming skills would not be assessed through direct observation as for science practical or spoken English skills, but would be based on the evidence they produced i.e. the program. If students worked collaboratively it could be difficult for teachers marking the NEA to determine each student's contribution. Therefore the potential for malpractice, whether accidental or deliberate, would remain.

### **5.2.2 Use coursework or a project to assess a clearly defined list of programming skills delivered over a set period (Option 5)**

Most of those who suggested students should demonstrate their programming skills in coursework or over a set period recognised that it would be prone to the same risks and opportunities for malpractice as the original NEA. As such, this proposal was considered only within the context of a separately reported grade, where malpractice in the programming task would not threaten the validity of the 9 to 1 grade. This proposal shares some of the weaknesses identified under option 4.

The approach would be similar to that used in other subjects which require students to complete a portfolio or lab book, and would record the skills which they obtained. Students could be rewarded for the achievements shown in their portfolio with a pass/fail grade. This judgement would either need to be made by teachers, monitored by the exam boards, or else entirely externally verified. This process could be burdensome to teachers.

A coursework or portfolio approach to the assessment of programming skills would still offer opportunities for malpractice. These risks would be mitigated, to some extent, if the tasks were set by teachers, as this might limit the extent to which worked solutions could be found online. This would, however, create extra work for teachers who would be required to set appropriate tasks that were sufficiently challenging and effectively target the skills to be assessed. If malpractice were to be widespread, the separately reported grade would not provide a reliable measure and would be of questionable value.

### **5.2.3 Exam boards issue progressively more difficult coding challenges during the two-year course, with students' competency judged by their teacher. Teachers and students agree which of these tasks to complete<sup>7</sup> (Option 6)**

Under this proposal, not all students would necessarily complete the same tasks, or tasks that were equally challenging. Some stakeholders suggested that this would be similar to the assessment of practical skills in A level science, with students receiving a pass/fail grade based on their teachers' judgements of their competency against a defined set of criteria. (The science tasks are not presented on any continuum of difficulty though, and students must be given the opportunity to undertake all of the practical tasks specified.) Some stakeholders would welcome the opportunity for teachers to offer differentiated programming tasks to students based on their ability. However, the optional nature of the tasks would create risks both in terms of potential malpractice and validity.

We can see no meaningful way to incorporate such an assessment into the qualification. The basis on which a pass/fail grade would be awarded is not clear, and it would be challenging to devise criteria which would allow teachers to make accurate judgements about the competency of students who had undertaken different tasks.

The approach might be more feasible if reported as a pass/merit/distinction grade similar to that for spoken language in GCSE English, with students being judged by their teachers against defined criteria on the competencies demonstrated in a series of tasks of escalating difficulty. However, under such a model, all students would need to attempt the same tasks in the same order (and at the same time, to limit opportunities for malpractice – although it is still possible that solutions would appear online shortly after the tasks were released). Students would need either to complete

---

<sup>7</sup> This proposal was distinct from the completion of specified programming tasks issued by the exam board and subsequently assessed in the examination (paragraph 5.3.2 below).

all tasks before the teacher reached a judgement about their competency, or opt to stop at a particular point and cap their attainment. This would remove much of the scaffolding and optionality identified by stakeholders as positive elements of such an approach.

Teachers' judgements would also need to be moderated in some way. It would likely also provide an additional challenge for teachers to manage classes where some students were continuing to undertake programming tasks, while others had chosen to stop.

#### **5.2.4 Use a collaborative programming project to assess programming (Option 7)**

Stakeholders who suggested this approach recognised a collaborative programming project would be vulnerable to the same opportunities for malpractice as the original NEA. We only considered this proposal, therefore, within the context of a separately reported - endorsed - grade for programming skills where malpractice in the programming task would not threaten the validity of the 9 to 1 grade.

This approach would require that the skills necessary to complete the project matched the specific content assessed by the exam papers. The exam boards or teachers could design the project, but an exam board set project would be more likely to ensure that students were equally prepared for their exams. Some stakeholders felt a collaborative approach to programming was preferable to an individual approach, as this would reflect the approach used in industry. However, to generate the endorsed grade, an individual student's work would have to be identifiable. Exam boards would likely require teachers to police student interactions to make sure each individual student's output was their own work. The lengths teachers would have to go to be assured they were assessing the work of only a specific student would seem unreasonable. As with all of the options involving a separately reported grade, any malpractice would undermine the reliability and value of the grade as a measure of programming ability.

It is also not clear how the assessment could take into account the extent to which group dynamics which might impact on an individual's ability to demonstrate their programming skills (for instance, where one student was prevented from being allowed effectively to contribute, or their contribution could not be identified). These challenges seem to undermine the feasibility of this approach.

## 5.3 Assess programming skills by examination

### 5.3.1 Assess programming skills using an online (completed using an internet browser) or on-screen (completed on a computer using a specific piece of software, but not necessarily while connected to the internet) task (Option 8)

Several respondents to our initial consultation, and many teachers we spoke to, argued that programming skills should be assessed electronically. This would replicate more closely the experience of programmers in industry. Electronic assessment was suggested as a solution to some of the potential opportunities for malpractice, particularly if completed under exam conditions or in an environment in which students' activity could be monitored.

Some exam boards already assess programming skills using on-screen assessment at A level and GCSE. Some teachers felt it would, therefore, be relatively straightforward for exam boards to 'borrow' from these existing assessment models.

Many other stakeholders we spoke to, however, felt that online (as distinct from on-screen) assessment would be too risky for the entire GCSE cohort. They were particularly concerned about the manageability and reliability of online assessment on this scale. Some felt schools' technology infrastructures vary too widely, so it would not be possible to ensure that all students would have the same experience, especially if they all had to complete the assessment at the same time. If an online assessment was not taken simultaneously then the same opportunities for malpractice would exist as were created by the original model.

Assessments which require candidates to be connected to the internet create additional risks of malpractice by allowing students to more easily discuss tasks and plagiarise code, compared to assessments which can be completed on-screen while network access is disabled. Some stakeholders suggested software created for online assessment of programming skills was particularly prone to errors in the form of bugs or outages which could have significant ramifications for students completing the assessment in limited examination time, and for the validity of the assessment overall. Certainly, exam boards would need to undertake substantive testing of any online assessments to mitigate these risks. This could significantly delay the implementation of such an approach.

For these reasons it was felt by most stakeholders that an on-screen (but not online) assessment of programming skills was a more feasible option. Some teachers suggested that it might be possible to adopt 'low tech' stop-gap solutions such as printing outputs or submitting them to the exam board on a CD or memory stick. They identified additional risks to such an approach associated with postage and file formats, but felt that these could be mitigated.

The current use of on-screen assessment by one exam board offering GCSE 9 to 1 computer science seems to demonstrate that this approach to the assessment of programming skills is feasible – at least for those schools and colleges that have chosen to enter their students with this exam board.

### **5.3.2 Students complete a range of programming tasks during their GCSE course with programming being assessed in targeted exam questions (Option 9)**

Stakeholders felt that this would be one of the most feasible and easiest of the options to implement, and allow teachers to tailor their teaching of programming skills to the individual needs of their students. While the tasks would be specified by the exam board, the skills developed by completing the tasks would be entirely assessed by the examination(s).

Some stakeholders were concerned that on the one hand, without controls such an activity could become a teacher demonstration, with students gaining little in the way of practical programming experience but on the other hand, that any such controls could place excessive burden on teachers. The recognised variation in teachers' subject knowledge within computer science could also lead to significant divergence in students' experiences. We would expect that the exam questions would be designed to avoid formulaic approaches that would be amenable to rote learning, and that students who had extensive exposure to practical programming would be better prepared for the exam than those who had not. We would evaluate the effectiveness of the questions used.

Stakeholders felt that a range of programming tasks would be easier to incorporate into regular classroom teaching than the 20 hours originally specified for the NEA and required under the interim approach. Some teachers also suggested that it might be possible to introduce this option for first teaching in 2019. If the programming skills were restricted to the second year of the course, this would allow the exam boards time to prepare and to release suitable sample assessment materials.

### **5.3.3 Students use pre-release materials to complete programming tasks and answer questions based on their experience in their exams (Option 10)**

Some stakeholders suggested students could use pre-release materials to undertake programming tasks, on which they would then be tested in the examination. Some pointed to International GCSEs in computer science that currently make use of this approach.

There are risks to the pre-release approach, however, primarily around the extent to which the questions asked in the exam may become predictable when teachers and students have seen a given piece of pre-release material. This in turn may encourage question-spotting which would have an impact on validity and narrow teaching and learning. Exam boards would need to take steps to mitigate these risks.

## **6. Our View**

Students taking GCSE computer science must develop an understanding of programming and programming skills, and these abilities must be assessed.

Having considered the options, we propose programming should be assessed by examination. The alternatives would place an unjustifiable and unsustainable burden on teachers, and could be vulnerable to malpractice.

Examinations must be sat by all students at the same time, under supervision, and set and marked by the exam board. Subject to these provisions, examinations can be taken on-screen (including online) and the use of pre-release materials can be designed in to the format of the assessment. The exam boards could provide tasks which teachers could use to support teaching and learning, but which would not formally be assessed.

In coming to this view, we have taken into account the malpractice in the legacy GCSE computing qualification and in the NEA in 2017.

In this subject, students' access to worked solutions and established industry working practices would likely compromise the validity of an individually awarded GCSE qualification that was assessed without the controls of an exam. Assessment by examination seems to be the only solution that mitigates the risks commensurate to the status of this qualification as a GCSE that is included in the EBacc performance measure.

There are several different assessment options that fall under our definition of assessment by examination, that exam boards could take.

We do not believe that any one of the options would necessarily be more valid than another. Our preferred option would mean that exam boards could choose how they effectively assess all of the required programming content in an exam context. We are keen to encourage innovation and GCSE computer science seems to offer particular opportunities for the exam boards. We do not wish to prescribe one specific approach to the assessment of programming skills that would prevent this from happening, beyond requiring that the assessment is taken under exam conditions.

We are also wary of stipulating a specific approach to assessment (for instance, online or on-screen assessment of programming skills) as we are mindful that not all schools and colleges have the necessary IT resources for all their students to sit such an assessment simultaneously<sup>8</sup>. We do not wish for our long-term approach to place additional pressures on centres' IT infrastructure, or to exclude students from taking a GCSE in the subject because their school or college could not support the assessment.

Granting exam boards greater discretion over the form of assessment they offer will enable them to take into account the varying IT resources within schools and colleges. The exam boards might take different approaches, catering for different levels of resource within centres.

Given that the current short-term arrangements for assessing programming skills appear to be working (albeit that teachers would like greater freedom to decide how to develop their students' programming skills) we can allow exam boards the time to develop, and teachers the time to prepare to deliver, revised versions of the qualifications which afford the most meaningful experience of programming skills to students.

---

<sup>8</sup> See the recent article in the *Times Educational Supplement*, [Exclusive: Third of school computers 'ineffective'](#) (TES, 25 September 2018)



## 7. Impact of Our Proposals

### 7.1 Equality Impact Assessment

When we [initially consulted on our proposals](#) to remove NEA from the qualification grade in GCSE computer science, we identified one particular group of students who might be adversely impacted by our eventual decision. These were certain disabled students whose disabilities meant they might perform less well in exams than in NEA. This includes students with visual impairments and those with disabilities that cause fatigue or difficulty concentrating during an exam. These students are entitled to reasonable adjustments to the way they take their exams.<sup>9</sup>

Assessment by examination may also pose issues for those students who, for reasons of extended medical treatment or other personal circumstances, are likely to be disadvantaged or unable to undertake all the assessment for the qualification at the time at which the examination is timetabled. Our Conditions require the exam boards to have in place clear arrangements for Special Consideration to be given to students who have temporarily experienced an illness, injury, or other event outside of their control which is likely to impact on their ability to take the assessment or to prevent them from demonstrating their true abilities.

We did not identify any other impacts on students who share protected characteristics. Some respondents to the original consultation expressed the view that our proposal would disadvantage female students who, they argued, perform better in non-exam assessment compared to boys, and worse in examinations. The evidence that we considered in the course of our [equality analysis for GCSE reform](#) (p.46), however, did not support this conclusion. We continue to be satisfied that the evidence shows that neither approach favours one gender over the other.

When we removed the NEA from the qualification grade, stakeholders suggested that certain pupils who possessed traits which might pre-dispose them to practical programming skills but who struggled to express their ideas in an exam would be negatively impacted. Dyslexic students were felt to be particularly likely to be affected. Permitting exam boards to adopt an approach which allows students to develop their skills through a more authentic programming activity (i.e. through requiring greater demonstration of practical programming skills) might have a positive impact on the experience of these students, and their performance in the qualification.

Revisiting the assessment objectives to place greater weight on AO3 ('Analyse problems in computational terms') rather than AO1 and AO2 may also improve the experience and outcomes of students who perform well in the practical programming aspects of the subject, but have difficulty expressing their ideas in writing.

Some stakeholders who responded to our previous consultation felt that GCSE computer science was likely to prove an attractive subject choice for some autistic students, who would have then found our subsequent changes to the structure of the qualification to be particularly distressing. At the time, this was balanced by the

---

<sup>9</sup> Further details are provided in our [Equality Impact Assessment for GCSE reform](#).

number of respondents who reported that the NEA had proved to be very stressful for some students. In particular, some autistic students had found the tasks to be challenging because certain aspects proved problematic for them to complete. These students may benefit from an approach where programming skills are assessed entirely by examination.

Allowing exam boards to be more innovative in their approach to the assessment of programming skills may also offer greater opportunities for the use of assistive technology by disabled candidates, including those assistive technologies which require internet connectivity to function effectively. This could potentially broaden the range of reasonable adjustments available to students in this subject, and could serve to demonstrate where they could be employed in other subjects in the future.

If the current short-term assessment arrangements are retained for an additional year, the exam boards will have time to develop and refine their approach and provide new sample assessment materials to centres in good time for teachers to prepare to deliver their courses of study. This means students will be able to make more informed decisions in Year 9 when they opt to study the subject.

We are inviting stakeholders who feel that there are other equalities impacts that we have not considered to make these known to us in their response to the consultation.

## 7.2 Regulatory Impact Assessment

It is difficult to estimate the regulatory impact of our proposals on centres and awarding organisations, as this will depend on the approach which each exam board adopts. It is particularly challenging given that we intend to afford exam boards the opportunity to innovate, and this may entail development costs or lead them to favour an approach to assessment which we have not considered.

There are, however, some impacts which would likely be common to any approach that assessed programming skills via examination, which we have considered.

### 7.2.1 Impact on centres

Some disruption and additional costs are likely to be experienced by centres who may choose this opportunity to switch exam board if the boards decide to adopt different approaches to assessing programming skills through an exam. As teachers' decisions will be informed by their judgements about the form of assessment they feel most comfortable preparing students for and which is best suited to the needs of their students, we feel that this limited negative impact is likely to be outweighed by the longer term benefits.

Our proposal would remove the requirement for teachers to set aside 20 hours for students to have the opportunity to undertake a programming task. Some stakeholders have suggested that the 20-hour requirement is burdensome because it takes up classroom time which could be better spent preparing students for the examination. However, the approaches which the exam boards may adopt to the assessment of programming skills may entail a similar amount of classroom preparation (for instance, time spent working on pre-release material) although this would be in preparation for assessments which would contribute to the overall qualification grade. Whether our proposal would be likely to lead to a net reduction in teacher burden would depend on the decisions made by each exam board. It is likely that the exam boards will consider the potential to differentiate between each other

on the basis of teacher burden when deciding on their preferred approach. Whatever approach they choose to adopt, our preferred option would give teachers greater flexibility to determine the best use of their teaching time by removing the requirement to set aside 20 hours for the programming task.

Depending on the approach to assessment by examination adopted by each of the exam boards, there may also be additional burden placed on schools due to the timetabling of any additional examinations or by the extension of existing exams to take into account the need to assess programming skills. It is reasonable to assume that the exam boards would seek to minimise the disruption that this would pose to centres by acting jointly through the Joint Council for Qualifications to take into account any changes in assessment duration or format when devising the exam timetable for 2022, when students sitting qualifications incorporating our proposed approach to the assessment of programming skills would first be assessed.

On balance, we feel that specifying an all examination approach to the assessment of programming skills is likely to be the least burdensome approach for teachers. Any approach which entails the reintroduction of the NEA would require internal teacher assessment or would mean unreasonable costs for the exam board. This might lead to an increase in fees for entering the qualification. Internal assessment would create burden for teachers in terms of marking and administration, and for the exam boards who would need to moderate that marking. Exam boards would also need to monitor to ensure that tasks were being completed within their rules. They would also have to investigate potential cases of malpractice, and apply sanctions against students, teachers and schools (as well as dealing with any appeals). Given the previous scale of malpractice in the computing and computer science qualifications at this level, these arrangements would need to be particularly robust in the initial years of the new approach. Providing the evidence to fulfil these monitoring requirements would also likely add to the administrative burden on teachers.

## 7.2.2 Implications for schools and colleges' resources

It is possible that exam boards may adopt an approach which places additional pressures upon a school's computer infrastructure. We are aware of recent reporting that up to [a third of computers in schools are 'ineffective'](#), and likely would not be suitable for the completion of online or on-screen assessment. However, given the potential for schools to change qualifications and the fact that the issues of resourcing in schools are well known, exam boards are likely to be mindful of this issue when determining their approach. Our proposal allows the exam boards the opportunity to take different approaches to assessing programming skills, so it is reasonable to assume that they will seek to cater for different levels of resourcing between schools. Exam boards have an obligation to ensure that their assessments are designed in such a way that centres are able to enter candidates for them.

On the other hand, we are aware that there are some schools who are already making extensive use of technology in their curriculum and within formative assessments. It may be the case that the current format of examinations, which are primarily still completed via pen and paper, is dissuading other schools from making greater use of technology within their teaching. There is the potential for more innovative approaches to the assessment of programming skills by the exam boards, such as online or on-screen assessments, to have a positive impact on computing

infrastructure in schools by removing some of the potential barriers to greater investment in (and use of) technology in the classroom.

### 7.2.3 Impact on exam boards

In principle, an all examination approach could lead to an increase in the costs incurred by the exam boards in comparison to the interim arrangements currently in place. However, these costs will be offset to some extent by the reduction in the costs to the exam boards who do not need to moderate NEA, or to conduct the additional monitoring to detect malpractice which we required when the qualifications were initially developed.

Beyond this, the regulatory impact on the exam boards is likely to depend on the approach they decide to adopt. It is possible that our approach would favour exam boards who have already adopted one of the approaches to assessment of programming skills which we discuss above, but all exam boards will need to revisit their assessment strategies if we were to re-weight the assessment objectives. Furthermore, all exam boards would be likely to want to reconsider their approach in light of our final decision.

### 7.2.4 Other considerations

This proposal takes into account the 2015 *Department for Education Protocol for changes to accountability, curriculum and qualifications*<sup>10</sup> to introduce minimum lead in times for significant changes and to do more to consider the impact on schools when introducing such changes.

We are inviting stakeholders who have additional intelligence about the likely regulatory impact of our proposals to make this known to us in their consultation response.

## 8. Consultation Questions

1. To what extent do you agree or disagree that assessment by examination is the best approach for assessing all of the content in GCSE computer science?

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please explain your answer.

2. To what extent do you agree or disagree that there is a range of ways by which programming could be assessed within our definition of an exam?

---

<sup>10</sup> [Department for Education Protocol for changes to accountability, curriculum and qualifications](#)

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please explain your answer, with reference to the options we have set out where appropriate.

3. If you do not support our proposed approach, how would you prefer programming skills to be assessed in GCSE computer science?

4. To what extent do you agree or disagree that the original assessment objective weightings should be reinstated?

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please explain your answer.

5. In our proposals we discuss the continued use of a statement signed by the head of centre confirming that students have been given the opportunity to design, write, test and refine programs, using one or more high-level programming language with a textual program definition as part of their course of study in preparation for the examinations. If programming skills are assessed in the exams, should such a statement be required?

Yes/No

Please explain your answer.

6. To what extent do you agree or disagree that the interim arrangements should remain in place for students who will take their exams in summer 2021, with the new assessment arrangements being used for the first time in summer 2022?

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please explain your answer.

7. We believe that our proposals will reduce the burden upon teachers, who will be able to decide how they develop their students' programming skills and how to timetable this in the course. To what extent do you agree or disagree?

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please explain your answer.

8. We do not believe there would be a negative impact from our proposals on any students because of their protected characteristics, beyond those already identified for whom reasonable adjustments can be made. To what extent do you agree or disagree?

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please explain your answer.

9. If you are responding on behalf of an exam board, please provide us with an indication of any additional costs you estimate you will incur if we implement our proposals as set out in this consultation.

In your response please distinguish between one-off and recurring costs.

10. We have a duty under the Apprenticeships, Skills, Children and Learning Act to have regard to the desirability of facilitating innovation in connection with the provision of regulated qualifications. We have committed in our Corporate Plan to survey awarding organisations' views of the impact of our regulatory requirements on innovation and consider any revisions required in response.

***We consider that our proposals are sufficiently flexible to allow exam boards to take innovative approaches to assessment in GCSE Computer Science, and do not believe that these proposals would unduly prevent innovation.***

To what extent do you agree or disagree with the above statement?

Strongly agree/ agree /neither agree nor disagree/ disagree/strongly disagree

Please provide specific examples to illustrate your answer.

11. Do you have any comments on the proposed changes to our rules for GCSE computer science presented in Annex A?

Yes/No

Please enter any comments below.

## Annex A: Proposed changes to our rules for GCSE computer science

We propose the following changes to our rules for GCSE computer science:

1. Remove existing rules that require students to complete a 20-hour programming project.
2. Change the weighting of the assessment objectives.
3. Introduce new rules that set out – at a high level – how exam boards must assess programming skills.

### I. Remove requirements for programming project

As set out in section 3, we propose that students beginning courses on or after 1 September 2020 will have their programming skills tested directly in their exams. This means they would not need to have dedicated time in their timetable to complete a programming project as part of their course of study. Similarly, exam

boards would not need to collect statements from schools and colleges to confirm they have set aside appropriate time for students to complete such a project.

To reflect this change, we propose to withdraw the following parts of our current *GCSE Subject Level Conditions and Requirements for Computer Science*<sup>11</sup> with effect for students starting their courses from 1 September 2020:

- Condition GCSE(Computer Science)3 – Programming Project.
- Requirements for the programming project.

## II. Revised assessment objective weightings

As we explain in section 3, we propose to revert back to the original assessment objective weightings for GCSE computer science. We will do this by simply changing the weightings set out in our rules.

## III. New rules for assessing programming skills

We also propose to introduce some new rules for exam boards on how they assess programming skills within the exams for GCSE computer science. We propose these rules will:

- require exam boards to assess students' ability to design, write, test and refine programs to a set task/specification (or to solve a problem).
- be clear that each of these skills can be assessed separately or in combination with any of the others and using single or multiple tasks.

Our proposed rules also include the existing requirements (currently included within requirements for the programming project) for:

- qualification specifications to set out which programming language(s) students can use in assessments.
- exam boards to justify their choice of permitted programming language(s) in their assessment strategy.

Our view is that these requirements should remain because qualification users need to understand which programming languages are covered by each specification, and because we need to understand how the different language(s) offered affect each exam board's approach to assessment.

## IV. Our proposed rules

We set out below our proposed revised rules in full. The new text we are proposing to add to the current rules is in **bold and underlined** type. Existing text we are proposing to remove is in ~~striketthrough~~ type.

### **GCSE Subject Level Conditions for Computer Science**

|  |   |
|--|---|
| <b>Condition GCSE(Computer Science)1</b> | <b>Compliance with content requirements</b> |
|--|---|

---

<sup>11</sup> [GCSE \(9 to 1\) subject level conditions and guidance for computer science](#)

|  |  |
|--|--|
| GCSE(Computer Science)1.1                | In respect of each GCSE Qualification in Computer Science which it makes available, or proposes to make available, an awarding organisation must –<br><br>(a) comply with the requirements relating to that qualification set out in the document published by the Secretary of State entitled ‘Computer science GCSE subject content’, <sup>12</sup> document reference DFE-00701-2014,<br>(b) have regard to any recommendations or guidelines relating to that qualification set out in that document, and<br>(c) interpret that document in accordance with any requirements, and having regard to any guidance, which may be published by Ofqual and revised from time to time. |
| GCSE(Computer Science)1.2                | In respect of each GCSE Qualification in Computer Science which it makes available, or proposes to make available, an awarding organisation must comply with any requirements, and have regard to any guidance, relating to the objectives to be met by any assessment for that qualification which may be published by Ofqual and revised from time to time.  |
| <b>Condition GCSE(Computer Science)2</b> | <b>Assessment<sup>13</sup></b>   |
| GCSE(Computer Science)2.1                | <del>[Not used]</del><br><br><b><u>An awarding organisation must ensure that in respect of each assessment for a GCSE Qualification in Computer Science which it makes available it complies with any requirements, and has regard to any guidance, which may be published by Ofqual and revised from time to time.</u></b>  |
| GCSE(Computer Science)2.2                | <del>[Not used]</del>  |
| GCSE(Computer Science)2.3                | <del>[Not used]</del>  |
| <b>Condition GCSE(Computer Science)3</b> | <b>Programming project</b>   |

---

<sup>12</sup> [Guidance: GCSE computer science](#)

<sup>13</sup> The numbering of paragraphs within this Condition has been adopted for consistency with previous versions of the Conditions; the proposed new Condition GCSE(Computer Science)2.3 is identical to the version of that Condition which was in force between 13 May 2015 and 15 January 2018.



- GCSE(Computer Science)3.1 In respect of each assessment cycle for a GCSE Qualification in Computer Science which it makes available, an awarding organisation must—
- (a) require each Centre to provide a programming project statement to the awarding organisation, and
  - (b) treat any failure by a Centre to provide a programming project statement to the awarding organisation in a timely manner as malpractice and/or maladministration (under General Condition A8 (*Malpractice and maladministration*)).
- GCSE(Computer Science)3.2 For the purposes of this condition, a ‘programming project statement’ is a true and accurate written statement made by a Centre to an awarding organisation which confirms that it has taken reasonable steps to secure that each Learner to which that Centre has delivered the assessments to be taken in a particular assessment cycle for a GCSE Qualification in Computer Science which the awarding organisation makes available—
- (a) has had the opportunity to undertake the programming project and has had 20 hours set aside in the timetable to allow them to undertake the project, and
  - (b) that their written accounts of their programming project represent their individual work, cover each part of the project and reference any resources used or support given
- GCSE(Computer Science)3.2 In respect of each GCSE Qualification in Computer Science which it makes available, or proposes to make available, an awarding organisation must comply with any requirements, and have regard to any guidance, relating to the objectives to be met by any assessment for that qualification which may be published by Ofqual and revised from time to time.

## Assessment objectives – GCSE Qualifications in Computer Science

Condition GCSE(Computer Science)1.2 allows us to specify requirements relating to the objectives to be met by any assessment for all GCSE Qualifications in Computer Science.

The assessment objectives set out below constitute requirements for the purposes of Condition GCSE(Computer Science)1.2. Awarding organisations must comply with

these requirements in relation to all GCSE Qualifications in Computer Science they make available or propose to make available.

|     | Objective   | Weighting                       |
|-----|---|---------------------------------|
| AO1 | Demonstrate knowledge and understanding of the key concepts and principles of computer science.                               | <del>35-40%</del><br><u>30%</u> |
| AO2 | Apply knowledge and understanding of key concepts and principles of computer science.   | <del>45-50%</del><br><u>40%</u> |
| AO3 | Analyse problems in computational terms:<br>to make reasoned judgements<br>to design, program, evaluate and refine solutions. | <del>15-20%</del><br><u>30%</u> |

## Requirements in relation to the programming project for GCSE Qualifications in Computer Science

Condition GCSE(Computer Science)3.3 allows us to specify requirements and guidance in relation to the programming project for GCSE Qualifications in Computer Science.

We set out our requirements for the purposes of Condition GCSE(Computer Science)3.3 below.

### Programming languages

For each GCSE Qualification in Computer Science which it makes available, or proposes to make available, an awarding organisation must ensure that—

- (a) it sets out in the specification for that qualification the programming language(s) which Learners are permitted to use for the purposes of the programming project,
- (b) each such programming language is a high-level programming language that has a textual program definition, and
- (c) it justifies its choice of permitted programming language(s) in its assessment strategy for the qualification.

### Programming Projects

An awarding organisation must ensure that each programming project—

- (a) is designed and set on the basis that it should be completed by each Learner during 20 hours set aside in the timetable,
- (b) is designed and set to be taken under conditions specified by the awarding organisation, including, in particular, conditions which support the Centre in ensuring that the evidence generated by each Learner can be Authenticated, and

- ~~(c) requires each Learner to undertake a single project which leads to the generation of the following evidence—~~
  - ~~(i) a program designed, written, tested and refined by the Learner, either to a specification or to solve a problem, using one or more of the programming languages specified by the awarding organisation, and~~
  - ~~(ii) a written report.~~

~~A programming project for a GCSE Qualification in Computer Science may be set—~~

- ~~(a) by the awarding organisation, or~~
- ~~(b) by a Centre.~~

~~In any event, the awarding organisation must demonstrate to Ofqual's satisfaction in its assessment strategy that—~~

- ~~(a) it has taken all reasonable steps to identify the risk of any Adverse Effect which may result from its approach to setting the programming project, and~~
- ~~(b) where such a risk is identified, it has taken all reasonable steps to prevent that Adverse Effect or, where it cannot be prevented, to mitigate that Adverse Effect.~~

## **Requirements in relation to assessments for GCSE Qualifications in Computer Science**

**Condition GCSE(Computer Science)2.1 allows us to specify requirements and guidance in relation to assessments for GCSE Qualifications in Computer Science.**

**We set out below our requirements for the purposes of Condition GCSE(Computer Science)2.1. Awarding organisations must comply with these requirements in relation to all GCSE Qualifications in Computer Science they make available.**

### **Programming languages**

**For each GCSE Qualification in Computer Science which it makes available, or proposes to make available, an awarding organisation must ensure that –**

- (a) it sets out in the specification for that qualification the programming language(s) which Learners are permitted to use for the purposes of the assessments,**
- (b) each such programming language is a high-level programming language that has a textual program definition, and**
- (c) it justifies its choice of permitted programming language(s) in its assessment strategy for the qualification.**

### **Assessment of Programming Skills**

**The subject content for GCSE Qualifications in Computer Science is set out in the document published by the Secretary of State entitled 'Computer Science**

**GCSE subject content', document reference DFE-00701-2014 (the 'Content Document').**

**Paragraph 5 of the Content Document states that GCSE Qualifications in Computer Science must –**

**... require students to develop the following skills:**

**[...]**

- 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**

**In designing and setting the assessments for a GCSE Qualification in Computer Science which it makes available, or proposes to make available, an awarding organisation must ensure that – taking those assessments together – Learners are required to –**

- (a) design,**
- (b) write,**
- (c) test, and**
- (d) refine,**

**a program to a set task/brief<sup>14</sup> (or to solve a problem), using one or more high-level programming language with a textual program definition (together, the 'Programming Skills').**

**For clarity, assessments may require Learners to demonstrate each of the Programming Skills –**

- (a) separately, or in any combination, and**
- (b) in relation to one or more separate tasks, briefs or problems.**

**An awarding organisation must demonstrate to Ofqual's satisfaction in its assessment strategy that –**

- (a) it has taken all reasonable steps to identify the risk of any Adverse Effect which may result from its approach to assessing Programming Skills, and**
- (b) where such a risk is identified, it has taken all reasonable steps to prevent that Adverse Effect or, where it cannot be prevented, to mitigate that Adverse Effect.**

---

<sup>14</sup> For clarity, we use the term 'brief' here to avoid possible confusion with the concept of a **qualification** specification. It should be interpreted as a synonym for the term 'specification' used in the Content Document.



## Annex B: Your data

### The identity of the data controller and contact details of our Data Protection Officer

This Privacy Notice is provided by The Office of Qualifications and Examinations Regulation (Ofqual). We are a 'controller' for the purposes of the General Data Protection Regulation (EU) 2016/679 and Data Protection Act 2018 ('Data Protection Laws'). We ask that you read this Privacy Notice carefully as it contains important information about our processing of consultation responses and your rights.

### How to contact us

If you have any questions about this Privacy Notice, how we handle your personal data, or want to exercise any of your rights, please contact:

Data Protection Officer at [dprequests@ofqual.gov.uk](mailto:dprequests@ofqual.gov.uk) or write to us at: Data Protection Officer, Ofqual, Earlsdon Park, 53-55 Butts Road, Coventry, CV1 3BH.

As part of this consultation process you are not required to provide your name or any personal information that will identify you however we are aware that some respondents may be happy to be contacted by Ofqual in relation to their response. If you or your organisation are happy to be contacted with regard to this consultation, please give your consent by providing your name and contact details in your response.

### Our legal basis for processing your personal data

For this consultation, we are relying upon your consent for processing personal data. You may withdraw your consent at any time by contacting us using the details above.

### How we will use your response

We will use your response to help us shape our policies and regulatory activity. If you provide your personal details, we may contact you in relation to your response.

### Sharing your response

We may share your response, in full, with The Department for Education (DfE) and The Institute for Apprenticeships (IFA) where the consultation is part of work involving those organisations. We may need to share responses with them to ensure that our approach aligns with the wider process. If we share a response, we will not include any personal data (if you have provided any). Where we have received a response to the consultation from an organisation, we will provide the DfE and IFA with the name of the organisation that has provided the response, although we will consider requests for confidentiality.

Following the end of the consultation, we will publish a summary of responses and may publish copies of responses on our website, [www.gov.uk/ofqual](http://www.gov.uk/ofqual). We will not include personal details.

We will also publish an annex to the consultation summary listing all organisations that responded. We will not include personal names or other contact details.

Please note that information in response to this consultation may be subject to release to the public or other parties in accordance with access to information law, primarily the Freedom of Information Act 2000 (FOIA). We have obligations to disclose information to particular recipients or including member of the public in certain circumstances. Your explanation of your reasons for requesting confidentiality for all or part of your response would help us balance requests for disclosure against any obligation of confidentiality. If we receive a request for the information that you have provided in your response to this consultation, we will take full account of your reasons for requesting confidentiality of your response, but we cannot guarantee that confidentiality can be maintained in all circumstances.

Members of the public are entitled to ask for information we hold under the Freedom of Information Act 2000. On such occasions, we will usually anonymise responses, or ask for consent from those who have responded, but please be aware that we cannot guarantee confidentiality.

If you choose 'No' in response to the question asking if you would like anything in your response to be kept confidential, we will be able to release the content of your response to the public, but we won't make your personal name and private contact details publicly available.

## How long will we keep your personal data

For this consultation, Ofqual will keep your personal data (if provided) for a period of 2 years after the close of the consultation.

## Your data

Your personal data:

- will not be sent outside of the European Economic Area
- will not be used for any automated decision making
- will be kept secure

We implement appropriate technical and organisational measures in order to protect your personal data against accidental or unlawful destruction, accidental loss or alteration, unauthorised disclosure or access and any other unlawful forms of processing.

Your rights, e.g. access, rectification, erasure

As a data subject, you have the legal right to:

- access personal data relating to you
- have all or some of your data deleted or corrected
- prevent your personal data being processed in some circumstances
- ask us to stop using your data, but keep it on record

If you would like to exercise your rights, please contact us using the details set out above.

We will respond to any rights that you exercise within a month of receiving your request, unless the request is particularly complex, in which case we will respond within 3 months.

Please note that exceptions apply to some of these rights which we will apply in accordance with the law.

You also have the right to lodge a complaint with the Information Commissioner (ICO) if you think we are not handling your data fairly or in accordance with the law. You can contact the ICO at [ico.org.uk](http://ico.org.uk), or telephone 0303 123 1113. ICO, Wycliffe House, Water Lane, Wilmslow, Cheshire SK9 5AF.

If there is any part of your response that you wish to remain confidential, please indicate so in your response.





© Crown Copyright 2019

This publication is licensed under the terms of the Open Government Licence v3.0 except where otherwise stated.

To view this licence, visit

[www.nationalarchives.gov.uk/doc/open-government-licence/](http://www.nationalarchives.gov.uk/doc/open-government-licence/)

or write to

Information Policy Team, The National Archives, Kew, London TW9 4DU

Published by:

**ofqual**

Earlsdon Park  
53-55 Butts Road  
Coventry  
CV1 3BH

0300 303 3344  
[public.enquiries@ofqual.gov.uk](mailto:public.enquiries@ofqual.gov.uk)  
[www.gov.uk/ofqual](http://www.gov.uk/ofqual)