Course report 2022

Subject	Computing Science
Level	Higher

This report provides information on candidates' performance. Teachers, lecturers and assessors may find it useful when preparing candidates for future assessment. The report is intended to be constructive and informative and to promote better understanding. It would be helpful to read this report in conjunction with the published assessment documents and marking instructions.

The statistics used in this report have been compiled before the completion of any appeals.

Grade boundary and statistical information

Statistical information: update on courses

Number of resulted entries in 2022	3490

Statistical information: performance of candidates

Distribution of course awards including grade boundaries

A	Percentage	36.0	Cumulative percentage	36.0	Number of candidates	1255	Minimum mark required	79
B	Percentage	18.2	Cumulative percentage	54.2	Number of candidates	640	Minimum mark required	66
C	Percentage	16.9	Cumulative percentage	71.1	Number of candidates	590	Minimum mark required	53
D	Percentage	13.6	Cumulative percentage	84.7	Number of candidates	475	Minimum mark required	40
No award	Percentage	15.3	Cumulative percentage	N/A	Number of candidates	535	Minimum mark required	N/A

You can read the general commentary on grade boundaries in appendix 1 of this report.

In this report:

- ‘most’ means greater than 70%
- ‘many’ means 50% to 69%
- ‘some’ means 25% to 49%
- ‘a few’ means less than 25%

You can find more statistical reports on the statistics page of SQA’s website.

Section 1: comments on the assessment

The 2021–22 course assessment provided candidates with a choice to complete either the ‘Database design and development’ or the ‘Web design and development’ option in both the question paper and assignment.

In the question paper, 54% of candidates completed the database design and development section and 46% of candidates completed the web design and development section. The average marks for each section were very similar.

In the assignment, 53% of candidates completed the database design and development task and 47% of candidates completed the web design and development task. The average mark for database design and development was higher than for web design and development.

Question paper

The question paper performed in line with expectations, with some questions proving to be more demanding than intended. Candidate performance was lower than what would be expected for Higher Computing Science. Feedback from the examining team and markers identified that there was a greater number of candidates who had not achieved the level of progression from National 5 than would normally be expected. Responses suggest that candidates struggled with questions that required extended writing in the form of an explanation.

The revised course assessment was initially introduced in 2019 and the lack of past papers available to exemplify the level of demand and style of questions for the full course content still impacts on performance. Question 6(e), where candidates were asked for a benefit of comprehensive testing, is a good example of this. This content is clearly stated in the Higher Computing Science Course Specification and in the appendices, yet very few candidates could give an answer relating to functional requirements, and instead their responses discussed types of test data.

The nature of software design and development questions with complex scenarios and stems requiring accuracy of reading, for example understanding that a shorter race time is better, or that fewer shots on a golf hole result in a win for that hole, also proved challenging for candidates.

Statistical evidence and feedback from the marking team suggest that questions 6(e), 7(c), 8(b), 8(c)(ii) and 8(d) proved to be more demanding than intended. This was taken into account when setting the grade boundaries.

Assignment

The assignment performed in line with expectations.

Feedback from markers, teachers and lecturers indicate that it was positively received and was fair and accessible for candidates. The majority of candidates understood what was required and completed two tasks in the allocated time.

No changes were made to grade boundaries in relation to the assignment.

Section 2: comments on candidate performance

Areas that candidates performed well in

Question paper

Software design and development, and computer systems

Question 2: Many candidates were able to access at least 2 of the 3 marks, but with some candidates incorrectly assigning a negative exponent.

Question 5: Many candidates were able to complete the missing values for the trace table and could subsequently correct the line of code.

Question 6(a): Many candidates examined the design to identify why it was not fit for purpose.

Question 6(b)(i): Most candidates could define a record structure for the scenario given in the stem.

Question 6(b)(ii): Many candidates could declare a suitable variable using the record structure that they defined in part (i).

Question 6(c): Many candidates could write code with good accuracy to increase speed for players with a playing time of over 500, using their record structure.

Question 8(a): Many candidates could analyse the given problem to identify two processes. However, candidates need to be careful not to use language associated with input or output.

Question 8(c)(i): Many candidates were able to use their programming language to substring for the first character of a string.

Database design and development

Question 9(b): Most candidates were able to draw an entity-relationship diagram for the given relational database tables.

Question 11(a): Most candidates were able to design a query using two tables with search criteria involving wildcards.

Question 11(b)(i): Most candidates were able to use the result of their previous query from part (i) to form a new query.

Question 12(c): Many candidates were able to write an SQL statement involving a calculation which included an equi-join.

Web design and development

Question 13: Many candidates could identify the labelled elements.

Question 14: Many candidates were able to accurately draw a wireframe for the given scenario which included indication of validation and a submit button.

Question 15(a): Many candidates could analyse the multi-level structure and explain why the navigation was not fit for purpose.

Question 15(c): Most candidates were able to use a grouping selector appropriately.

Areas that candidates found demanding

Question paper

Software design and development, and computer systems

Question 3: Most candidates were not able to compare vector and bit-mapped applications for making the alteration specified in the stem and instead, resorted to stating a characteristic of vector or bitmap graphic.

Question 4: Many candidates were not able to consider the problem-solving aspect of this question relating to size of cache, and stated generic facts about the presence of cache memory compared to not having cache.

Question 6(e): Few candidates were able to describe a benefit of a comprehensive test plan, focusing on test data rather than functional requirements.

Question 7(a): Few candidates were able to analyse the scenario to identify boundaries and most resorted to re-stating parts of the stem of the question.

Question 7(d)(i): Many candidates were not able to identify the mismatch in the number of actual and formal parameters.

Question 7(d)(ii): Leading on from 7(d)(i), many candidates were not able to correctly call the function with the correct actual parameter.

Question 8(b): Most candidates were not able to design a read from file into a suitable data structure, despite this being an often-undertaken practical task and being present in every assignment task and signposted in the revision support for candidates.

Question 8(c)(ii): Although many candidates could identify a substring involving the first character, few candidates were able to identify a substring which involved a variable as a parameter.

Question 8(d): Many candidates were not able to cope with the level of problem-solving to implement placing a non-duplicate character into the array. Some marks in this question were relatively straight forward, such as traversing the array and the IF statement. Many candidates who attempted this question were rewarded with partial marks, even if the full solution was incorrect.

Database design and development

Question 12(d): Many candidates were not able to form the correct syntax of an UPDATE command.

Web design and development

Question 15(d): Many candidates were unable to identify the effect of the CSS rules on the HTML lists, and their answers lacked detail of which colours applied to which list items.

Areas that candidates performed well in or found demanding

Assignment

Software design and development

Task 1(a): Many candidates were not able to provide two assumptions. Many candidates repeated the facts provided in the stem of the question without conducting a closer analysis of the input.

Task 1(b): Many candidates were able to design the function to convert the user's inputs to strings that start with a capital letter.

Task 1(c): Most candidates were able to implement a modular program with appropriate procedures. Most candidates followed the supplied program design and made correct use of a record data structure with appropriate parameter passing.

Task 1(d): Many candidates were not able to describe the use of a watchpoint in the context of their own code. Some candidates confused a watchpoint with a breakpoint.

Task 1(e): Many candidates were not able to make appropriate comments on efficiency and maintainability in the context of their modular program. Some candidates incorrectly described fitness for purpose whilst others provided generic readability comments that did not extend beyond National 5 standard.

Database design and development

Task 2(a): Most candidates were able to complete an entity occurrence diagram.

Task 2(b): Most candidates were able to implement the SQL statement requiring an aggregate function, an equi-join, a ‘group by’ and an ‘order by’.

Task 2(c): Most candidates were able to implement the SQL statement(s) requiring an aggregate function, an equi-join, a ‘group by’ and two queries.

Task 2(d): Most candidates were able to re-write the query to include a wildcard.

Task 2(e): Most candidates were able to identify the functional requirement that could not be met.

Web design and development

Task 3(a): Many candidates were able to create end-user and functional requirements

Task 3(b): Most candidates were able to design a wireframe.

Task 3(c): Most candidates were able to implement HTML, CSS and Javascript to hide and display sections.

Task 3(d): Most candidates were able to implement HTML and CSS to float images and text to match the design.

Task 3(e): Many candidates were not able to provide appropriate testing examples, instead they offered generic testing definitions that did not extend beyond National 5 standard.

Section 3: preparing candidates for future assessment

Question paper

Centres should continue to refer to the Higher Computing Science Course Specification and cover all content to ensure that candidates are familiar with less frequently sampled areas.

Centres should encourage candidates to be proficient in the use of correct terminology and vocabulary for the subject and use the terminology accurately in explanations. ‘Explain’ type questions typically involve an element of problem-solving set out in the stem, and candidates should be encouraged to read and decode the stem and not merely make generic statements.

In questions relating to analysis, candidates should be clear about what is meant by functional requirements and should not express these in the form of user requirements. Candidates should also be made aware of the differences between inputs, outputs and processes, and that boundaries require some analysis of the problem and are not simply a restatement of the problem.

In software design and development, it is important to note the distinction between design and implementation. Candidates should be encouraged to recognise the difference between design in the form of pseudocode and that of implementation in the form of a programming language. Candidates often answer design questions using code, and this is accepted as pseudocode — even if the language used is SQA Reference Language.

However, in questions requiring a programming language, candidates should attempt to answer an implementation question in a single programming language of their choice, not in a hybrid of two or more languages. They should be made aware that the strict syntax of the language is not required.

Higher tariff design and programming language questions typically have a mixture of C and A marks. Candidates should be encouraged to attempt such questions, because even incomplete or not fully correct or working solutions can still be awarded marks.

Areas of challenge in software design and development include:

- accuracy of language in stating inputs, processes, outputs and the boundaries of a problem
- data flow in design
- parameter passing and using parameters correctly in the call of a function or procedure
- expressing evaluation in context for the given solution

In database design and development, candidates should be aware that:

- functional requirements should be a feature implemented by the database, for example through a query which includes features such as aggregate functions or calculations
- although candidates are accurate in their use of the syntax of the SELECT statement, candidates should be equally familiar with the syntax of other SQL statements, particularly UPDATE and DELETE as these will be used in conjunction with a wildcard in the Higher question paper

In web design and development, candidates should be aware that:

- wireframes for a form should include both validation appropriate to the problem specified and a submit button
- an element is a technical term, meaning NAV is a suitable response and not ‘nav bar’ or ‘navigation’
- a description of the application of CSS rules to a section of code should be detailed as to which exact HTML element is being affected

Assignment

While most centres deliver the course content detailed in the Higher Computing Science Course Specification, a few centres have deviated from this content when teaching practical implementation. The Course Specification exists to ensure consistent and transparent assessment year on year. Marking instructions are designed to assess the course content. Candidates are at risk of not being able to access all available marks for a question if they use techniques or constructs not specified in the Course Specification.

All standard algorithms should be implemented as refined steps of code and should not use inbuilt features of the software. Centres should adhere to the list of SQL operations, HTML, CSS and Javascript code provided.

Centres should remind candidates to follow the top-level design provided when implementing the software design and development task. Internal commentary should be in the context of the program being developed and, while essential, is not required for every line of code.

Centres should ensure that candidates appreciate the value of the analysis, design, testing and evaluation stages of the development process in the assignment. Candidates should complete these sections in the context of the task and to a standard that is appropriate for Higher level. Many candidates provided generic or National 5 level responses.

Appendix 1: general commentary on grade boundaries

SQA's main aim when setting grade boundaries is to be fair to candidates across all subjects and levels and maintain comparable standards across the years, even as arrangements evolve and change.

For most National Courses, SQA aims to set examinations and other external assessments and create marking instructions that allow:

- a competent candidate to score a minimum of 50% of the available marks (the notional grade C boundary)
- a well-prepared, very competent candidate to score at least 70% of the available marks (the notional grade A boundary)

It is very challenging to get the standard on target every year, in every subject at every level. Therefore, SQA holds a grade boundary meeting for each course to bring together all the information available (statistical and qualitative) and to make final decisions on grade boundaries based on this information. Members of SQA's Executive Management Team normally chair these meetings.

Principal assessors utilise their subject expertise to evaluate the performance of the assessment and propose suitable grade boundaries based on the full range of evidence. SQA can adjust the grade boundaries as a result of the discussion at these meetings. This allows the pass rate to be unaffected in circumstances where there is evidence that the question paper or other assessment has been more, or less, difficult than usual.

- The grade boundaries can be adjusted downwards if there is evidence that the question paper or other assessment has been more difficult than usual.
- The grade boundaries can be adjusted upwards if there is evidence that the question paper or other assessment has been less difficult than usual.
- Where levels of difficulty are comparable to previous years, similar grade boundaries are maintained.

Grade boundaries from question papers in the same subject at the same level tend to be marginally different year on year. This is because the specific questions, and the mix of questions, are different and this has an impact on candidate performance.

This year, a package of support measures including assessment modifications and revision support, was introduced to support candidates as they returned to formal national exams and other forms of external assessment. This was designed to address the ongoing disruption to learning and teaching that young people have experienced as a result of the COVID-19 pandemic. In addition, SQA adopted a more generous approach to grading for National 5, Higher and Advanced Higher courses than it would do in a normal exam year, to help ensure fairness for candidates while maintaining standards. This is in recognition of the fact that those preparing for and sitting exams have done so in very different circumstances from those who sat exams in 2019.

The key difference this year is that decisions about where the grade boundaries have been set have also been influenced, where necessary and where appropriate, by the unique circumstances in 2022. On a course-by-course basis, SQA has determined grade boundaries in a way that is fair to candidates, taking into account how the assessment (exams and coursework) has functioned and the impact of assessment modifications and revision support.

The grade boundaries used in 2022 relate to the specific experience of this year's cohort and should not be used by centres if these assessments are used in the future for exam preparation.

For full details of the approach please refer to the National Qualifications 2022 Awarding — Methodology Report.