Course report 2022

Subject	Computing Science
Level	National 5

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	6440

Statistical information: performance of candidates
Distribution of course awards including grade boundaries

A	Percentage	40.9	Cumulative percentage	40.9	Number of candidates	2635	Minimum mark required	84
B	Percentage	20.6	Cumulative percentage	61.5	Number of candidates	1325	Minimum mark required	70
C	Percentage	16.4	Cumulative percentage	77.9	Number of candidates	1060	Minimum mark required	57
D	Percentage	12.4	Cumulative percentage	90.3	Number of candidates	795	Minimum mark required	43
No award	Percentage	9.7	Cumulative percentage	N/A	Number of candidates	625	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 both the question paper and assignment, 49% of candidates completed ‘Database design and development’ and 51% of candidates completed ‘Web design and development’. The average marks for each option were very similar.

Some candidates completed both optional topics, with the highest mark taken forward to the total mark.

Evidence from markers and statistical analysis suggest that the question paper was more demanding for candidates than intended, particularly coding questions 8(c)(ii) and 10(c). Higher tariff design and coding questions are designed to assess problem solving skills and contain a mixture of more accessible C marks, and more challenging A marks. Candidates generally scored lower on these types of questions than expected, often struggling to obtain the C or A marks appropriate to their ability. Many candidates missed question 14(a) on identifying key attributes in the ‘Database design and development’ section. Question 19(e) on end-user requirements in the ‘Web design and development’ section also proved more demanding than intended.

The distribution of marks awarded shows that the assignment performed as intended. Analysis and evaluation questions continue to be the most challenging tasks for candidates. In the ‘Software design and development’ task, the design question was more challenging than expected, however this was balanced by the programming question being less challenging than in previous years, likely due to the detailed pseudocode provided in the design. Candidates continue to score highly in the coding questions in all three tasks, with most candidates making very good attempts at these questions.

Grade boundaries were adjusted downwards on account of the question paper. No adjustment was made relating 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 1:	Most candidates were able to convert decimal to binary.
Question 2:	Many candidates could use a programming language of their choice to display the output given, using text and variables.
Question 3:	Most candidates were clear on identifying programming variable types.
Question 4(a):	Most candidates could read a graphical design to identify a loop.
Question 4(b):	Many candidates could apply a design technique to a problem.
Question 5:	Most candidates could identify the mantissa and exponent from a given floating-point representation.
Question 6(b):	Many candidates could identify a logical operator used.
Question 7:	Most candidates could identify objects and their attributes from a given design.
Question 8(a):	Many candidates could read a problem and then identify the processes required to produce a solution.
Question 8(c)(i):	Many candidates could identify the data structure and the data type required to be used in a given problem.
Question 8(d):	Most candidates could answer how information could be transferred securely.
Question 9(a)(i):	Most candidates were knowledgeable of graphical design techniques.
Question 9(a)(ii):	Many candidates could explain why the design stage could be revisited.
Question 9(b)(ii):	Many candidates could identify test data for a given problem.
Question 9(c)(ii):	Many candidates could describe how altering the variables would make the code more readable.
Question 9(c)(iii):	Many candidates could calculate the number of bits required.
Question 9(c)(iv):	Many candidates could state the type of error.

Question 9(d): Many candidates could identify the type of translation applied.

Question 10(a): Many candidates could design a user interface using the information provided in the problem, and clearly show the inputs and output.

Question 10(b)(i): Most candidates could apply a computer systems question to an implementation question on calculation.

Question 10(b)(ii): Many candidates could apply a computer systems question to an implementation question on temporary storage.

Question 10(d)(i): Many candidates could describe how bit mapped graphics are stored.

Question 10(d)(ii): Most candidates could describe how energy consumption could be reduced.

Database design and development

Question 11: Many candidates could identify the database data type.

Question 12(a): Many candidates could read the given SQL code to provide the expected output.

Question 13(a): Most candidates could show the relationship between the two entities by using the information given.

Question 13(b)(i): Most candidates could design a query from the information provided.

Question 13(b)(ii): Many candidates could complete the SQL for the problem given.

Question 14(a)(i): Most candidates who answered this question were able to identify the key attributes on the diagram, however some candidates did not read the instruction above the question.

Question 14(a)(ii): Most candidates could identify the required attribute from the Task entity by using the functional requirements and the ERD.

Question 14(b): Many candidates understood the GDPR and its application in this problem.

Question 14(c): Many candidates demonstrated their ability to write SQL when provided with a problem.

Question 14(d): Most candidates could write efficient SQL code from code provided.

Web design and development

Question 15: Many candidates could identify a property and class from CSS code provided.

Question 16: Most candidates could identify the type of link.

Question 17: Most candidates could identify why the wireframe provided did not meet the functional requirements provided.

Question 18(b)(i): Most candidates could read the HTML and the CSS provided to state the colour of the heading when the webpage is displayed.

Question 18(b)(ii): Many candidates could write a single style rule.

Question 18(c): Many candidates could apply the information given to provide the code for displaying the graphic.

Question 18(d)(i): Many candidates could identify a file type to be used with the problem given.

Question 19(a)(i): Many candidates could draw how the web page would look in a browser.

Question 19(a)(ii): Many candidates could complete the missing line of code.

Question 19(b): Many candidates could state the purpose of the HTML code provided.

Question 19(c): Most candidates could identify the computer language used to implement the problem.

Areas that candidates found demanding

Question paper

Software design and development, and computer systems

Question 6(a): Many candidates were not able to decompose the condition in line 5 to provide the correct input value.

Question 8(b): Many candidates were able to pick up some of the more accessible marks by identifying that a loop was required with an input statement. Only some candidates were able to fully understand the problem and gain full marks for a complete working design.

Question 8(c)(ii): Some candidates were able to identify that a random number was required to select a seat, but very few candidates were able to use the random number within a loop to identify an empty seat in the array and then update the seat to be occupied. Some candidates responded using a graphical design technique which does not allow them to access any marks in a question asking for code in programming language.

Question 8(c)(iii): Many candidates were able to demonstrate that they understood that more full seats required a random number to be generated more often until an empty seat was found. However, few candidates were able to discuss this in terms of efficiency using appropriate computing terms such as the use of loops and random numbers.

Question 9(b)(i): Most candidates were able to identify that a loop and input statement was required, however only some candidates could apply the full input validation standard algorithm. Candidates must be prepared to apply a standard algorithm to an unfamiliar context.

Question 9(c)(i): Few candidates were able to explain that the indentation was for the IF statement, giving generic answers (for example referencing loops) rather than applying their knowledge to the context of the question.

Question 10(c): Many candidates were able to identify that a loop with an input statement was required to replace 7 to 11. Some candidates identified that the running total algorithm was required. Few candidates used an array with an index inside the loop to gain maximum marks.

Database design and development

Question 12(b): Most candidates were not able to describe that the code could be tested by comparing the expected results against the actual results.

Web design and development

Question 18(a)(i): Most candidates incorrectly identified the design as a wireframe. Rather than a low-fidelity prototype.

Question 18(a)(ii): Most candidates were able to state that the design in part (i) shows how the completed web page would look. Few could provide a second reason, such as to test navigation and showing the prototype to a client.

Question 18(d)(ii): Most candidates could not give a reason for the large file size that related to the graphic provided in the question. Candidates need to apply their knowledge to the context of the question.

Question 19(d): Most candidates were unable to provide the full name of the law that had been broken. Candidates should be encouraged not to abbreviate their answers.

Question 19(e): Most candidates were unable to provide a clear reason as to why the feedback would become an end-user requirement. Instead, many candidates referred to the auction.

Areas that candidates performed well in or found demanding

Assignment

Software design and development

Task 1(a): A good variety of solutions was observed. Many candidates failed to achieve the mark for storing the output message, instead indicating the message should be output immediately.

Task 1(b): Candidates followed the design well, with the majority coding a running program. Some candidates lost marks in the output section of the design. Often the five weights or messages were not stored earlier, or the average weight was not rounded correctly.

Task 1(c): The majority of candidates completed the test table correctly and provided output from their working program.

Task 1(d): Many candidates failed to relate their answer to their own code and therefore provided no evaluation. To gain these marks, candidates had to justify their answers through reference to their own code or by providing specific examples of lines of code. General, rote answers for efficiency, robustness and readability do not allow candidates to access these marks.

Database design and development

Task 2(a): The attributes of the house entity were correctly identified by the majority of candidates. The mark for identifying the unique value proved harder to achieve.

Task 2(b): Most candidates successfully identified the primary and foreign keys.

Task 2(c): The majority of candidates successfully wrote correct, working SQL statements. Some candidates forgot to join the two tables in part (c)(iii).

Task 2(d): Most candidates successfully identified the errors in the SQL statement provided.

Task 2(e) Some candidates failed to relate fitness for purpose to the functional requirement. Some candidates did not make a definitive yes or no statement and struggled to justify their answer.

Web design and development

Task 3(a):	Some candidates struggled with this task as they did not understand the difference between end-user and functional requirements.
Task 3(b):	Most candidates provided comprehensive wireframe designs. Where full marks were not achieved, it was often because candidates did not include the consistent content from other pages of the website.
Tasks 3(c),(d),(e):	Most candidates coded the required HTML content well, but some found the CSS edits required more challenging. Some candidates were not able to access all marks as they had not coded everything required in the task.
Task 3(f):	Some candidates failed to relate fitness for purpose to the functional requirements. Some candidates did not make a definitive yes or no statement and struggled to justify their answer. This is also an opportunity for candidates to reflect on any requirements in the previous tasks that they know they have not been able to code.

Section 3: preparing candidates for future assessment

Question paper

Candidates should be encouraged to read the questions carefully and look at the number of marks assigned to the question. For ‘describe’ and ‘explain’ questions, this will guide them as to how many points they need to make (usually either one or two).

Teachers and lecturers should aim to further develop candidates in problem-solving skills and application of the standard algorithms for both ‘design’ and ‘write code’ questions.

Candidates should read the 3, 4 and 5-mark questions carefully to:

- make sure that they answer using a ‘design technique’ or ‘programming language of their choice’, as required by the question
- use any variable names given in the question
- identify which standard algorithm they need to apply

Where candidates are given a design and asked to write code in a programming language of their choice, they should follow the design. This is similar to following the more complex designs given in the assignment.

Candidates should ensure they know the different design methodologies and purpose of each. In ‘Web design and development’, many candidates did not understand the difference between wireframes and low-fidelity prototypes.

Candidates should be prepared to identify end-user requirements and functional requirements. To do so they must understand the difference between them.

Assignment

Teachers and lecturers should ensure candidates can identify end-user and functional requirements in all three areas. They should also ensure that candidates can then evaluate where these requirements have been successfully met at the end of a task or project.

Candidates should be provided with opportunities to evaluate their own program code, providing examples of fitness for purpose, efficient use of coding constructs, robustness, and readability. Feedback should focus on candidates’ discussion of their own code in their evaluations. In general, rote answers for efficiency, robustness and readability do not allow candidates to access marks in evaluation.

Candidates should be encouraged to provide a definitive ‘yes’ or ‘no’ answer and justify this when asked to evaluate fitness for purpose in ‘Database design and development’ and ‘Web design and development’. In some tasks, there is no definitive answer, but marks are awarded if the answer and justification is consistent with the candidate’s solution.

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.