Course report 2026

National 5 Computing Science, SCQF level 5

We've written this report for teachers and lecturers. We recommend you read this with the published course assessment documents and marking instructions.

The report:

- provides a summary of how each course component performed
- highlights areas of the question paper and coursework where candidates performed well and areas where they did not perform well
- has advice you might find useful when preparing candidates for assessment

We collect and prepare the statistics in course reports before we complete the appeals process.

Copyright © Qualifications Scotland 2026

Statistical and grade boundary information

Course entries

Number of resulted entries in 2025: 6,583

Number of resulted entries in 2026: 6,444

Distribution of course awards (including minimum mark to achieve each grade)

Course award	Number of candidates	Percentage	Cumulative percentage	Minimum mark required
A	2,669	41.4	41.4	84
B	1,229	19.1	60.5	71
C	984	15.3	75.8	58
D	740	11.5	87.2	45
No award	822	12.8	100	Not applicable

We've not applied rounding to these statistics.

In this report:

- 'Most' means greater than or equal to 70%.
- 'Many' means 50% to 69%.
- 'Some' means 25% to 49%.
- 'Few' or 'a few' means less than 25%.

You can find statistical reports on our website.

General commentary on grade boundaries

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

For most National Courses, we aim 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's very challenging to get the standard on target every year, in every subject, at every level. Therefore, we hold a grade boundary meeting for each course to bring together all the information available (statistical and qualitative) and make final decisions on grade boundaries based on this information. Senior Qualifications Scotland staff normally chair these meetings.

Principal assessors use their subject expertise to evaluate the performance of the assessment and propose suitable grade boundaries based on the full range of evidence. We 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.

- We can adjust the grade boundaries downwards if there's evidence that the question paper or other assessment has been more difficult than usual.
- We can adjust the grade boundaries upwards if there's evidence that the question paper or other assessment has been less difficult than usual.
- If levels of difficulty are comparable to previous years, we maintain similar grade boundaries.

Every year, we evaluate the performance of our assessments in a fair way, while ensuring standards are maintained so that our qualifications remain credible. To do

this, we measure evidence of candidates' knowledge and skills against the national standard.

For full details of the approach, please refer to the Awarding and Grading for National Courses Policy.

How the course assessment performed

Question paper and assignment

This year, approximately 63% of candidates completed the ‘Database design and development’ sections, and 38% of candidates completed the ‘Web design and development’ sections in both the question paper and the assignment.

The question paper performed as intended and candidates performed well in it. The higher tariff problem-solving questions were effective in differentiating candidates. Candidates who attempted these questions were able to gain some of the marks with partially correct or incomplete solutions.

The question paper performed as intended.

In general, the assignment performed as intended, with implementation continuing to be an area of strength across all topic areas.

However, feedback and analysis on statistical data indicated that the ‘software design and development’ design task (task 1b) was more challenging than intended. Responses to this task suggested that several candidates gave responses that did not address the requirements of the task, which prevented them from accessing the marks designed to be ‘C’ grade marks. We adjusted the grade boundaries to account for this.

How candidates performed

Question paper

Section 1: Software design and development, and computer systems

Data representation

Data representation continues to be an area in which most candidates perform well and consistently gain marks. Candidates demonstrated a solid understanding of binary number representation, with most converting a denary number to binary (question 6(a)). Most candidates also performed well in floating-point representation, identifying the mantissa and exponent from a given floating-point value (question 10(c)). Candidates successfully applied their understanding of graphical representation. Most candidates identified an object in a vector graphic (question 6(c)(i)) and described how bit-mapped graphics are represented in a computer systems memory, using binary values for individual pixels (question 6(c)(ii)).

Computing structure, environmental impact and security precautions

Performance on questions relating to computer structure, environmental impact and security precautions was mixed. While many candidates demonstrated an awareness of energy consumption and methods to reduce the environmental impact of computers, some candidates provided methods that did not apply while the computers were in use (question 1). In addition, although many candidates understood the role that firewalls play in keeping data secure, some incorrectly referred to data being read by others during transfer (question 3).

Candidates were more successful with short questions assessing direct recall of basic computer architecture and translation. Most candidates identified the role registers play in temporarily storing data during calculations (question 9(c)). Many candidates demonstrated an understanding of the role the data bus during the transfer of data (question 11(c)) and identified complier as the appropriate type of translator (question 11(g)).

Analysis and design

Candidates performed well on questions assessing analysis and design. In the analysis question (question 8(a)), many candidates gave two processes from the given scenario.

The user interface was a particular area of strength, with most candidates successfully designing an interface for user instruction, input, and the output the program would produce (question 10(a)(i) and (ii)). Many candidates also applied selection and decision-making logic, creating a design that would determine a character's status based on the rules provided (question 11(a)).

Many candidates continue to struggle with recognising and producing a design that requires input validation using conditional loops. Candidates who did not use a conditional loop were unable to access any of the 4 marks (question 9(a)). Candidates also made errors when using arithmetic expressions and assignment to refine a step involving a calculation. Their designs often omitted one of the required calculations or did not store the result correctly (question 9(b)).

A significant number of candidates missed the straightforward mark available for question 5 because they named the design technique rather than identifying that the design used a conditional loop.

Implementation

Responses varied across the cohort to questions assessing candidates understanding of programming constructs. Most candidates correctly identified the logical operator in the code provided (question 4(a)). Candidates showed a good understanding of loops, with many identifying the required loop and providing an explanation that related to the scenario of the question (question 8(b)). While most candidates identified the need for an array of integers in question 10(b)(i), fewer candidates identified the correct data types in question 8(d), with float and Boolean often given for cost and refreshment.

Candidates' application of programming constructs when writing code was more consistent. Many candidates identified the appropriate pre-defined function to process data from the scenario and successfully wrote the corresponding code, using

round in question 2, length in question 8(c) and random in question 11(f). Candidates formatted an output statement successfully, with most gaining both marks for combining text and variable into a single display message (question 11(d)).

As with previous years, questions requiring candidates to use the running total standard algorithm within a loop were particularly challenging (question 10(b)(ii)). This question had the highest no response rate in section 1. Some candidates did gain full marks, but many who attempted this question did not recognise that the required data was already stored within the array, or they omitted indexing the array.

Testing

Testing-related questions were challenging.

Most candidates did not provide a clear description of the purpose of normal test data, so did not gain the available mark (question 4(b)(i)). Many candidates also did not provide a complete example of normal test data, and provided test data for only half of the given condition (question 4(b)(ii)).

Candidates confused syntax, execution, and logic errors, with many incorrectly identifying the error as either a logic or syntax error and few correctly identifying it as an execution error. However, candidates more often identified that the error resulted from division by zero (question 11(b)).

Section 2: Database design and development

Design

Questions assessing database design concepts and skills continues to be an area that generates positive performance. Most candidates correctly identified key attributes in an entity-relationship diagram, evidencing their knowledge of how records can be uniquely identified within a database (question 13). A greater number of candidates correctly provided appropriate validation for fields based on the sample data (question 14). Many candidates also explained why the text data type was appropriate for the data being stored (question 15(a)). Query design was also an area of strength, with many candidates correctly identifying the fields, tables, search criteria and sort order to retrieve the specified data (question 15(b)(i)).

In contrast, few candidates gained the mark for applying their knowledge of GDPR in the context of the question. A common incorrect response was that the data had not been kept secure, whereas the breach related to the inappropriate use of data or sharing data without permission (question 15(b)(ii)).

Implementation and testing

Candidate performance in SQL-related questions was relatively strong.

Many candidates successfully constructed an INSERT, SELECT and UPDATE statement using the correct values from the question stem (questions 15(c), 16(a) and 16(b)). In general, candidates also identified appropriate fields, applied search criteria and used a join to retrieve data from multiple tables, demonstrating a sound understanding of SQL query construction (questions 16(a) and 16(b)).

Candidates also demonstrated good debugging skills, with many evaluating and explaining why the SQL statement would not remove all the required records. Even more candidates then rewrote the SQL statement to remove all the correct records (questions 12(a) and (b)).

Explaining, using the context of the question, why referential integrity had been broken, was particularly challenging for candidates (question 15(d)). Most candidates did not gain the mark for this question, which also had the highest non-response rate in the paper.

Section 3: Web design and development

Design

Wireframing and the application of content layout, labelling and links based on the information provided were areas of strength, with most candidates gaining all three marks available for question 19. Candidates weren’t as secure in their understanding of audio file formats and their benefits, specifically MP3 and WAV (question 21(c)), with few candidates gaining full marks.

Implementation

Interpreting HTML and CSS code to determine how webpage elements would be displayed was an area in which most candidates performed well. Most candidates interpreted the effect of supplied code on webpage presentation, including alignment, list formatting, font styles, text size and background colour (questions 20(b) and 21(a)).

Candidates' performance was less consistent when asked to write HTML code. While many candidates used a title tag appropriately (question 20(a)) and image tag to change what was displayed in browser (question 20(c)(ii)), using the appropriate syntax for the HTML anchor tag and folder path was more challenging (questions 20(c)(i) and 20(c)(ii)).

Similarly, performance varied when candidates were required to describe or explain the purpose of a stylesheet, why a stylesheet would be used, and why a class was used (questions 17(a),17(b) and 18). Responses to these questions were often vague. However, most candidates demonstrated a good understanding of event-driven functionality and correctly identified that JavaScript and the onmouseover event had been used to implement an interactive web feature (questions 20(d)(i) and (ii)).

Assignment

Software design and development

Analysis, implementation and testing

Candidates continue to demonstrate strong analytical skills, with most identifying at least one additional process that the program would carry out to meet the required functionality (task 1a). Many candidates managed to gain full marks.

Coding a solution using a problem description and given design also remains an area of strength for candidates, with most candidates implementing a valid solution. The average mark for the implementation task remained stable. However, some candidates did not identify that the program required input within a conditional loop.

In addition, some candidates incorrectly used a single if statement for input validation rather than a conditional loop (task 1c).

Many candidates, regardless of whether they had a working solution, applied normal test data to determine the expected output (task 1d). Some candidates used their program to help them when testing, by comparing results. The blank answer line at the top was intended for candidates to record the error message produced after entering invalid data. A first review of candidate responses confirmed the feedback teachers provided that this part of the task did not perform as intended. We therefore disregarded any responses on this answer line and re-allocated the mark within the 2 marks available for testing.

Design and evaluation

Task 1b was intended to assess candidates' ability to design an algorithm to traverse a 1D array. Many candidates treated the task as an input validation design, as in several previous assignments, or did not identify that the output required a loop to display multiple values. Some candidates also designed the entire program, which the task did not require. As these difficulties stemmed from the wording of the task, we adjusted the grade boundaries to mitigate the impact.

Evaluating code which they have implemented continues to be an area that candidates struggle with. Some candidates provided appropriate evaluations relating to the readability of their code. However, despite repeated guidance, many candidates gave generic, rote-learned responses about fitness for purpose or efficiency without referring to specific examples from their own solution (task 1e).

Database design and development

Analysis and design

Candidates performed well when identifying input requirements (task 2b). In contrast, performance was weaker when creating functional requirements from a given scenario (task 2a). Most candidates attempted to write functional requirements, but many responses did not relate these requirements to searching the database.

Performance in database design was good. Most candidates gained 1 mark by interpreting the database analysis and writing appropriate range and restricted-choice validation rules to maintain data integrity (task 2c). However, some candidates only partially completed the validation required in the data dictionary. Candidates should check each field against the supplied information in turn.

Implementation and testing

Candidate performance when implementing and debugging SQL statements remained strong. Most candidates demonstrated a good understanding of data retrieval and data entry, successfully completing SELECT and INSERT statements using appropriate fields, tables and values (task 2d(i) and (ii)). As expected, writing an SQL statement that required more complex search conditions, like specifying a range of dates, was more challenging for candidates (task 2d(iii)).

Most candidates debugged a given SQL statement, allowing them to identify an issue with the WHERE clause, ORDER BY clause, or a missing join (task 2e).

Web design and development

Analysis and design

Candidates who completed the web option performed better when creating functional requirements based on a given scenario, with many gaining the 2 marks available (task 2a). Candidates who did not gain both marks often identified a requirement, but expressed these as end-user requirements instead. Consistent with the question paper, wireframing and understanding of layout and organisation were areas of strength, and most candidates successfully drew a wireframe that incorporated the required text and media elements (task 2b).

Implementation and evaluation

Most candidates implemented the HTML structure for the given design. However, performance in CSS implementation was less consistent. While many candidates implemented the required styles for the page headings, some candidates did not apply the correct CSS syntax or use the partial id styles supplied for the top, voteUs,

and bottom sections (task 3c). Candidates should ensure that they spend time identifying the purpose of any supplied styles before editing a supplied stylesheet.

Candidates' ability to apply their knowledge of JavaScript event-driven functionality to the context of the task and their code varied. Many candidates gained the mark available, but some gave a generic use of JavaScript in web pages instead of relating their answer to the web page they had just implemented (task 3d). Slightly more candidates were successful in evaluating the fitness for purpose of their web page against the stated functional requirements (task 3e).

Preparing candidates for assessment

You should encourage candidates to spend more time reading questions or tasks carefully, and avoid making assumptions about what is required. This affected candidate performance significantly this year (questions 1, 5 and 15(b)(ii) in the question paper, and task 1b and 2a in the assignment).

Candidates must answer according to the command word, for example, ‘identify’, ‘describe’ or ‘explain’. Questions to describe or explain often require candidates to apply their knowledge to the context of the question. This also applies in the assignment, where candidates are required to ‘evaluate’ their own code. Rote-learned answers that do not refer to the candidate’s own code will not gain a mark.

Encourage candidates to spend more time reading and interpreting any designs supplied in the tasks. When candidates complete tasks, they should refer back to the design to check that their solution matches the requirements.

Although this continues to be an area of improvement, you should continue to develop candidates’ problem-solving skills and prepare them to apply the standard algorithms to unfamiliar contexts for both ‘design’ and ‘write code’ questions in the question paper, and in design and implementation tasks in the assignment.

You should encourage candidates to read questions carefully to:

- determine which standard algorithm they need to use in either design or implementation
- consider how they will adapt the standard algorithm they have learned to the context of the question

You should also ensure that candidates can design and implement input validation using a conditional loop.