Course report 2025

National 5 Computing Science

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. You should read the report with the published assessment documents and marking instructions.

We compiled the statistics in this report before we completed the 2025 appeals process.

Grade boundary and statistical information

Statistical information: update on courses

Number of resulted entries in 2024: 6,744

Number of resulted entries in 2025: 6,583

Statistical information: performance of candidates

Distribution of course awards including minimum mark to achieve each grade

Course award	Number of candidates	Percentage	Cumulative percentage	Minimum mark required
A	3,279	49.8	49.8	86
B	1,192	18.1	67.9	74
C	926	14.1	82.0	62
D	613	9.3	91.3	50
No award	573	8.7	100%	Not applicable

We have not applied rounding to these statistics.

You can read the general commentary on grade boundaries in the appendix.

In this report:

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

You can find statistical reports on the statistics and information page of our website.

Section 1: comments on the assessment

In the question paper, candidate performance improved compared to previous years. This improvement was evident in the higher tariff problem-solving questions, as well as the more predictable short response questions.

The higher tariff problem-solving questions proved to be effective in differentiating candidates, with A candidates accessing full marks and C candidates accessing some marks. There were significantly fewer question papers with no response across the paper as a whole, and particularly in the larger problem-solving questions, which is encouraging. This indicates that C-grade candidates are attempting these questions and accessing some of the marks with a partially correct or complete solutions.

Overall, the question paper was deemed to have been slightly less demanding than intended due to the increased number of predictable, 1-mark, short response questions.

The assignment performed as intended, with candidates accessing more marks in practical coding tasks than the written tasks assessing analysis and evaluation skills.

This year, approximately 60% of candidates completed the ‘Database design and development’ sections, and 40% completed the ‘Web design and development’ sections.

We adjusted the grade boundaries at grades A and C due to the question paper being less demanding than intended.

Section 2: comments on candidate performance

Areas that candidates performed well in

Question paper

Software design and development, and computer systems

Most candidates performed well in the 1-mark questions on computer systems from the following areas of the course:

- question 1(a): data representation
- question 1(b): converting denary to binary
- question 4: identifying the mantissa and exponent
- question 5: graphic representation
- questions 9(c)(i) and 9(c)(ii): basic computer architecture
- question 9(d): environmental impact
- question 11(f): security precautions

In software design and development, candidates performed well in short-response questions in which they were asked to ‘state’ or ‘identify’, for example:

- questions 3(a) and 3(b) and questions 11(c)(i) and 11(c)(ii): answering questions about program code provided
- question 7(a): stating suitable data types
- question 8: answering questions about the design provided
- question 10(a): identifying processes
- question 10(c): completing the test table

In the larger design and implementation questions, candidates’ performance improved compared with previous years.

Most candidates were able to design a user interface in question 11(g), and most candidates were able to access some marks in the standard algorithms in questions 10(b) and 11(a). More candidates made an attempt at these types of questions than in previous years and were often able to access at least one mark.

Database design and development

Candidates performed well in questions related to database design.

Most candidates were able to identify key attributes and relationships between the entities in the entity-relationship diagram in question 15(a) and to state a requirement of the company under the UK General Data Protection Regulation (GDPR) in question 15(c).

In question 16(a)(i), most candidates could design a query for the problem given.

In implementation, many candidates correctly completed the SQL statement to apply the correct sort in question 13, and the search in question 15(b).

Many candidates were able to read and understand the given SQL code to provide the expected output in question 16(b) and explain why an error message was displayed in question 16(d).

Web design and development

In design, many candidates could state one reason why JPEG files are used on a website in question 20(a); most candidates could draw the navigation structure for the website described in question 21(a); and many candidates could identify the Copyright, Design and Patents Act 1988 in question 21(d)(ii).

Many candidates could write the CSS code for a heading in question 21(b) and could read and understand the HTML and CSS code to describe how the heading would look in question 21(e).

Many candidates were also able to answer the short response questions, for example, to identify the JavaScript event in question 17 and creating functional requirements in question 19.

Most candidates were able to state why the web page in question 21(d)(ii) was not fit for purpose.

Areas that candidates found demanding

Question paper

Software design and development, and computer systems

Candidates appear to find questions that require them to ‘describe’ or ‘explain’ computing concepts and computational behaviour demanding.

This was observed where candidates were asked to describe a situation during the software development process in question 2, and to describe concatenation (or simply to use the term ‘concatenation’ in their answer) in question 6.

More generally, candidate responses to ‘describe’ and ‘explain’ questions tend to be poorly expressed, do not use the appropriate technical language or fail to relate to the context of the question.

Questions 12 (a), (b) and (c) were designed to be challenging, and candidates found these to be demanding.

While there was a significant improvement in the higher tariff problem-solving questions (questions 9(b), 10(b) and 11(a)), many candidates still struggle to apply problem-solving skills and provide a solution to the problem given.

Database design and development

Some candidates found writing the SQL statement to update the database in question 16(c) to be demanding. Candidates either did not write the correct overall structure for an update statement or failed to include both the values being updated.

Web design and development

Candidates found questions asking them to ‘describe’ or ‘explain’ to be challenging, for example describing how file sizes could be reduced when applied to sound in question 21(c)(ii) and explaining the difference between absolute and relative addressing in question 18.

Areas that candidates performed well in or found demanding

Assignment

Software design and development

Most candidates were able to:

- task 1a: identify processes
- task 1b: design an input validation using length check
- task 1d: follow the given program design and write working code
- task 1e: evaluate the readability of their code

Many candidates failed to use the text for the entered fruits shown in the program design in their expected output (task 1c).

Some candidates still wrote an input validation using a single if statement rather than a conditional loop. This was evident in both the design (task 1b) and implementation (task 1d) stages.

Many candidates still fail to refer to their own code when writing an evaluation. Generic statements on code efficiency and robustness resulted in the failure to access two of the three evaluation marks (task 1e).

Database design and development

Most candidates were able to:

- task 2b: identify the required database inputs
- task 2c: identify the foreign key in the data dictionary
- task 2d: successfully code and run SQL insert, delete and select statements
- task 2e: identify the errors in the given SQL

Some candidates were able to identify a requirement from the given information, but failed to then create a functional requirement (task 2a). Candidates' answers were often expressed as end-user requirements.

Web design and development

Most candidates were able to:

- task 3b: write HTML code to implement headings, divisions, hyperlinks and correctly sized images
- task 3c: describe a suitable test for their web page
- task 3d: explain a use for JavaScript
- task 3e: draw an alternative navigation structure

Many candidates were able to:

- task 3b: write CSS rules to implement colours, fonts, text sizes and link colours

Some candidates were able to identify a requirement from the given information but failed to then create a functional requirement (task 3a). These answers were often expressed as end-user requirements instead.

While most candidates were able to implement CSS rules, many rules were not fully complete (task 3b). For example, the candidate may have coloured the element and applied the font successfully, but the omitted the font size, and therefore failed to achieve the mark.

Section 3: preparing candidates for future assessment

Teachers and lecturers should encourage candidates to read the questions carefully. Candidates must answer according to the command word, for example, 'identify', 'describe' or 'explain'. In particular, questions that require candidates to 'describe' or 'explain' will often require them to apply their knowledge to the context of the question, and they will not access marks with rote answers. This also applies in the assignment, where candidates are required to 'evaluate' their own code.

Teachers and lecturers 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 design and implementation tasks in the assignment. They should encourage candidates to read the 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

Teachers and lecturers should also ensure candidates can design and implement an input validation using a conditional loop.

For further support with these areas, refer to the candidate evidence and commentaries from the 2024 National 5 Computing Science question paper, which can be found on the N5 Computing Science Understanding Standards web page.

In the 'Database development and design' and 'Web development and design' sections, teachers and lecturers should ensure that candidates understand the language required to express the difference between end-user and functional requirements.

Once candidates have completed their assignment, teachers and lecturers should support candidates in ensuring that only pages containing candidate evidence that

will be marked is submitted to SQA. Any pages containing only instructions must be discarded. Evidence should also be submitted in the order of the task.

Appendix: 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 is 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 to make final decisions on grade boundaries based on this information. Members of our 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. 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.

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.

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.