Course report 2024

Higher 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 2024 appeals process.

Grade boundary and statistical information

Statistical information: update on courses

Number of resulted entries in 2023: 3,562

Number of resulted entries in 2024: 3,746

Statistical information: performance of candidates

Distribution of course awards including minimum mark to achieve each grade

A	Number of candidates	1,437	Percentage	38.4	Cumulative percentage	38.4	Minimum mark required	84
B	Number of candidates	630	Percentage	16.8	Cumulative percentage	55.2	Minimum mark required	72
C	Number of candidates	656	Percentage	17.5	Cumulative percentage	72.7	Minimum mark required	60
D	Number of candidates	488	Percentage	13.0	Cumulative percentage	85.7	Minimum mark required	48
No award	Number of candidates	535	Percentage	14.3	Cumulative percentage	100	Minimum mark required	N/A

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 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

Feedback from markers, teachers and lecturers indicated that the question paper and assignment were positively received, fair and accessible for candidates. This was supported by analysis on statistical data that demonstrated that both assessments performed as intended. As a result, no changes were made to grade boundaries. In the question paper it was noted that candidates appear to be improving at writing all forms of code, but questions requiring an extended response using keywords ‘describe’ or ‘explain’ remain the most challenging.

In the question paper, 64% of candidates completed the ‘Database design and development’ section, and 36% completed the ‘Web design and development’ section. In the assignment, 63% of candidates completed the ‘Database design and development’ section, and 37% completed the ‘Web design and development’ section.

Section 2: comments on candidate performance

Areas that candidates performed well in

Question paper

Software design and development, and computer systems

Question 1(a)(b): Most candidates were proficient in converting from denary to 8-bit two’s complement, and many candidates could identify the upper end of the range.

Question 3(a): Most candidates gained two or three marks when converting to floating-point representation.

Question 4: The majority of candidates made a good attempt at finding the longest string in an array, but candidates should be encouraged to use a design technique.

Question 6(b): Many candidates could correctly call a function using assignment and the actual parameter.

Question 8(d): Most candidates could identify actual and formal parameters from the given code.

Question 9(d)(ii): While most candidates appeared to struggle to accurately describe what is meant by a resource starvation DOS attack in part (i), most candidates could identify a cost associated with a DOS attack.

Question 10(a): Many candidates could complete the trace table, gaining 3 or 4 marks.

Question 10(c): Many candidates could name a debugging technique but seemed to find the description a little more challenging.

Question 10(d): Most candidates could explain the scope of a local variable.

Database design and development

Question 11(a): Most candidates could identify a functional requirement and its associated aggregate function.

Question 11(b): Most candidates could complete an entity-occurrence diagram with the appropriate entity and instance names, and correct associations between instances.

Question 13(b): Many candidates wrote an SQL DELETE statement using a wildcard.

Question 13(c): Many candidates demonstrated they could design an SQL statement that required a calculation, aggregate function and the use of grouping.

Question 14(b)(i): Most candidates could provide the expected output of an SQL statement consisting of an aggregate function, grouping and a double sort.

Question 14(c): Most candidates could write an SQL statement requiring an aggregate function, alias and search criteria with a logical OR operator.

Web design and development

Question 15: Most candidates wrote efficient CSS code that made use of grouping selectors as instructed.

Question 16(a): Many candidates demonstrated that they understood that personas are fictitious users created to represent users of the website.

Question 18(b)(i): Many candidates could accurately draw a wireframe for the given scenario.

Question 18(d): Most candidates knew that compatibility testing should consist of testing to check display and functions were as intended on different browsers and different devices.

Areas that candidates found demanding

Question paper

Software design and development, and computer systems

Question 2: Only some candidates could articulate the comparison between agile and iterative methodologies using relevant concepts and vocabulary.

Question 8(a): Only some candidates could analyse the scenario and identify two boundaries. Many responses were simply a restatement of the question stem.

Question 8(c): Only some candidates demonstrated critical thinking by examining the code and then articulating a response using appropriate programming terminology.

Question 9(d)(i): Very few candidates described what is meant by resource starvation DOS attack, with many omitting any mention of a resource such as a processor, RAM or backing storage.

Database design and development

Question 12: Only some candidates identified that tournamentID formed part of a compound key, and so could not be left blank.

Question 14(b)(ii): Only some candidates explained why the GROUP BY command was required in SQL statement to produce the expected output they provided in part (i).

Web design and development

Question 17(a)(ii): Only some candidates provided all the CSS code to style the horizontal navigational bar as instructed.

Question 17(b)(i): Only some candidates identified that the call to the function was missing or wrote the line of code to call the function with the appropriate event.

Areas that candidates performed well in or found demanding

Assignment

Software design and development

Task 1(c): The implementation of the program continues to be an area of strength for candidates. Most candidates implemented a modular program with appropriate function, procedures and standard algorithms. Most candidates followed the supplied program data flow, refinements and correctly used three parallel arrays with appropriate parameter passing.

Task 1(d)(i): Some candidates did not notice references to ‘sample test data’ and responded by stating that ‘Selop’ was not in the csv file. Some candidates noted that two companies had the same salary but did not explain why this would result in incorrect output.

Task 1(d)(ii): Only some candidates described additional refinements.

Database design and development

Task 2(a): Most candidates created functional requirements from a list of end-user requirements.

Task 2(b)(ii): Most candidates correctly identified the compound key required for the Result entity.

Task 2(c): Most candidates implemented the SQL statement requiring an Alias, a COUNT aggregate function, equi-joins, search criteria and GROUP BY clause.

Task 2(d): Most candidates accessed 3 marks. However, the fourth mark proved difficult to access as the condition used in part 1 of the query had to be repeated in part 2 to eliminate any swimmers in other lanes with the same time. Many candidates did not consider this.

Task 2(e): Most candidates identified the errors and rewrote the SQL statement with appropriate search criteria and ORDER BY clause.

Task 2(f): Only some candidates made sufficient reference to the database structure, which should include reference to both the field and the table.

Web design and development

Task 3(a): Most candidates designed a multi-level navigation structure.

Task 3(b): Most candidates implemented HTML and JavaScript to hide and display the bedroom descriptions.

Task 3(e): Most candidates reviewed the website and stated two reasons why it was not fit for purpose.

Section 3: preparing candidates for future assessment

Question paper

Candidates appear to have become much more familiar with the level of demand of questions and the standard of response required. There is continued improvement in areas such as floating-point representation, writing code for both parallel arrays and arrays of records in the context of the different algorithms, identification of parameters and the calling of functions. Trace tables is also an area where there has been an improvement. In design, there has been some improvement in identifying data flow, but this remains an area where there is still room for improvement.

Understanding code in an unfamiliar context and being able to answer 'explain' type questions about code is one of the more demanding areas of the course. Centres should consider learning and teaching strategies to address this, emphasising that candidates should be using the correct terminology and vocabulary of the subject.

Candidates were weaker on both analysis and evaluations. For example, in analysis, when attempting to identify functional requirements or boundaries, candidates often resorted to restating the stem of the question, while evaluation responses often lacked context. Centres should ensure candidates appreciate the value of the analysis and evaluation stages of the development process, and that they are able to provide responses in the context of the question and to a standard that is appropriate for Higher.

Centres should also practise extended response questions requiring description or explanation for all topics of the course. The focus should be on improving accuracy of expression and using appropriate technical language.

Assignment

While most teachers and lecturers continue to deliver the course content detailed in the Higher Computing Science Course Specification, some deviate 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 or task if they use techniques or constructs that are 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. Teachers and lecturers should adhere to the list of SQL operations, HTML, CSS and JavaScript code provided.

The analysis stage of the database design and development option continues to improve. Candidates should be encouraged to use the words 'A query to …' when writing functional requirements. This will prompt the use of appropriate query terminology, for example, search, sort and calculate, and may help eliminate incorrect references to end-user requirements.

It is usual in the assignment for a single mark to be awarded for an answer that has multiple parts, for example, 1 mark for all the inputs, 1 mark for data flow IN and OUT of a module, or

1 mark for each column of a test table. Candidates should be encouraged to focus on careful analysis of the questions and not make assumptions based on mark allocation.

Expected query output or completed web page illustrations are provided for support, but candidates should still conduct careful analysis and write meaningful code to correctly produce the output. The marking instructions are applied to use of appropriate constructs in the code, and the marker uses the output for reference only. For example, in task 2(c), some candidates grouped by initial. While this resulted in a match with the expected output in a table with no duplicate initials, this would not be considered a reliable GROUP BY option if applied to a larger data sample. In task 3(c), some candidates achieved the correct visual output for the footer but did so by creating extremely large margins, which is not appropriate HTML code. Example output will continue to be provided in the assignment, but candidates should be encouraged to consider whether their solution is robust.

Appendix: 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.

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.

During the pandemic, we modified National Qualifications course assessments, for example we removed elements of coursework. We kept these modifications in place until the 2022–23 session. The education community agreed that retaining the modifications for longer than this could have a detrimental impact on learning and progression to the next stage of education, employment or training. After discussions with candidates, teachers, lecturers, parents, carers and others, we returned to full course assessment for the 2023–24 session.

SQA's approach to awarding was announced in March 2024 and explained that any impact on candidates completing coursework for the first time, as part of their SQA assessments, would be considered in our grading decisions and incorporated into our well-established grading processes. This provides fairness and safeguards for candidates and helps to provide assurances across the wider education community as we return to established awarding.

Our approach to awarding is broadly aligned to other nations of the UK that have returned to normal grading arrangements.

For full details of the approach, please refer to the National Qualifications 2024 Awarding — Methodology Report.