Course report 2019

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 post-results services.

Section 1: comments on the assessment

Question paper

The question paper largely performed as expected. Feedback from the marking team and teachers and lecturers suggested it was fair in terms of course coverage. However, analysis of the statistical data indicated that questions 17(d)(i) and 18(a) were both more demanding than expected for the C grade candidates. The grade boundaries at C level were adjusted to take these questions into account.

Assignment

This was the second year since the assignment changed from an internally assessed bank of tasks to a single assignment, issued annually and externally assessed by SQA.

The assignment has three tasks, one for each practical area of the course:

- software design and development
- database design and development
- web design and development

The assignment largely performed as expected, with candidates achieving the anticipated range of marks on most tasks. The average overall mark rose slightly.

The SQL delete task 1(c)(ii) was less challenging than intended, with the majority of candidates achieving full marks. The software evaluation task 2(e) was more challenging than expected with few candidates achieving all four marks. Taking both tasks into account, no change was made to grade boundaries in relation to the assignment.

Section 2: comments on candidate performance

Areas that candidates performed well in

Question paper

Section 1:

Question 1: The majority of candidates demonstrated that they could confidently carry out an 8-bit binary to decimal conversion.

Question 2: The majority of candidates could correctly insert the given HTML elements to the code provided in the question.

Question 3: The majority of candidates were able to design an appropriate user interface for the given problem.

Question 4: Many candidates identified the manufacturer ASC sort, however many candidates did not identify the productCode DESC sort. Care needs to be taken by candidates in this question to ensure they accurately identify the fields.

Question 5(c): Most candidates could explain why the design did not have to be changed.

Question 6: Half of candidates were able to identify that the website was not fit for purpose as it did not meet the problem requirements.

Question 7(a): Over half of candidates identified the extreme values for the pair of dice. Some candidates only identified extreme values for a single die.

Question 8(a): The majority of candidates could identify a suitable type of validation to be used.

Question 8(b): Most candidates could explain why the SQL statement did not produce the intended results.

Question 9: The majority of candidates who attempted this question answered it well. However some candidates did not provide a bullet point list with the three items in the correct order and left aligned.

Question 10(a): Over half of candidates correctly identified the correct attribute type.

Question 10(b): The majority of candidates could identify the part of the processor where the condition would be evaluated.

Question 11: The majority of candidates correctly identified syntax as the type of programming error, however several did not correctly describe the effect of this error.

Question 12: The majority of candidates could identify the attributes of the object.

Section 2

Question 13(a): The majority of candidates could describe one process carried out by the program, just less than half could describe the second process.

Question 13(b)(ii): Over half of candidates could state a standard code for representing text characters and the number of bits used to store each character.

Question 13(c): Many candidates could correctly identify the data types to be used.

Question 13(d): Most candidates could identify the mantissa and exponent.

Question 13(f): Several candidates did not attempt this question, of those that did, most correctly coded a solution in a language of their choice.

Question 14(b): Most candidates were able to pick up some marks for this question. Some candidates who addressed each bullet in the question were able to achieve full marks.

Question 14(c): Many candidates could identify the type of validation, but several gave a generic, rote learned description, rather than applying it to the given scenario.

Question 14(d): Most candidates were knowledgeable about the implications of the GDPR but several did not relate their answer to the youth club.

Question 15(b)(i): The majority of candidates could describe what the company must do to avoid prosecution under the Copyright, Design and Patents Act.

Question 15(c)(i): The majority of candidates could identify the JavaScript event from the HTML code provided.

Question 15(c)(ii): The majority of candidates could describe the purpose of the code.

Question 15(d): The majority of candidates were confident in security precautions.

Question 16(a): The majority of candidates could correctly identify the design technique.

Question 16(d)(i): The majority of candidates could correctly identify the data bus.

Question 17(a): The majority of candidates could explain what had to be done to ensure that the startDate was not left blank.

Question 17(c)(i): The majority of candidates could explain why the SQL statement was not fit for purpose.

Question 17(c)(ii): Most candidates could re-write the SQL statement.

Question 18(a): Most candidates could exemplify the website structure with a home page, four linked multimedia pages and an external link. The majority of candidates either scored 0 or 4 marks suggesting that the full range of marks was less accessible for C grade candidates than intended. Some candidates answered using a wire frame or a low-fidelity prototype rather than using a navigational structure.

Question 18(b)(ii): Many candidates could complete part or all of the style rule.

Question 18d(i): The majority of candidates identified the correct graphic file format.

Question 19(c)(ii): Most candidates identified array as the correct data structure and integer as the correct data type.

Assignment

Task 1(a): Most candidates performed very well in this task and gained full marks.
Markers' feedback stated that some candidates did not gain the mark for identifying that the number of seconds stored in 'duration' should be a number and not a time.

Task 1(b): Most candidates clearly demonstrated that they could assign validation to fields. However, markers' feedback stated that some candidates set a maximum field size of 6 rather than setting the field length to equal 6.

Task 1(c)(i): Most candidates were capable of writing and implementing an SQL query across two tables, however many candidates did not implement the required join.

Task 1(c)(ii): Most candidates successfully deleted the required field using SQL.

Task 2(b): Candidates performed well in the programming task. Candidates were unable to access particular marks where they failed to implement the given design.

Task 2(c): Most candidates successfully identified suitable examples of test data.

Task 2(d): Most candidates who supplied a test run for their program gained a mark for this task.

Task 3(b): Most candidates successfully created the required web page with internal CSS and all the page content identified in the design. Some candidates did not identify that the three headings should be styled with a single CSS rule. In addition, some candidates did not correctly divide and colour the page.

Task 3(c): Most candidates identified at least one test that could be performed on their web page. Some candidates included hyperlinks as a test, despite the page having no links.

Task 3(d): Candidates who successfully related their answer back to the functional requirements gained a mark in this task.

Areas that candidates found demanding

Question paper

Section 1:

Question 5(a): The majority of candidates could not identify that a fixed (unconditional) loop was used in the design.

Question 5(b): The majority of candidates could not state the total displayed. Many provided the working but not the total, as required by the question.

Question 7(b) Few candidates identified the standard algorithm. This standard algorithm is provided on page 48 of the course specification.

Section 2

Question 13(b)(i): The majority of candidates were unable to describe how bit-mapped graphics are stored.

Question 13(e): Several candidates did not attempt this question. Of those that did, many did not correctly compete line 18 by providing the end of the loop (conditional or pre-conditional) and the condition.

Question 14(a): Several candidates could not identify functional requirements from the information provided in the question. Candidates tended to confuse functional requirements with user requirements.

Question 15(a): Several candidates could not identify end-user requirements from the information provided in the question. Candidates tended to confuse user requirements with functional requirements.

Question 15(b)(ii) : Many candidates could not state the advantage of one file type compared to the other.

Question 15(c)(iii): Many candidates could not explain why the code provided should be included in each page of the website. This question was assessing their understanding of why CSS is used.

Question 16(b)(i): Few candidates correctly stated the condition used in the loop construct.

Question 16(b)(ii): A significant number of candidates did not attempt the question. Of those who did, the majority could not identify the programming constructs used.

Question 16(c): Several candidates did not attempt this question. Of those who did, most achieved marks for the loop and opening the door.

Question 16(d)(ii): Several candidates could not explain how data is stored in memory.

Question 16(e): The majority of candidates could not explain why this particular program was compiled.

Question 17(b): Candidates could identify the tables required, but often did not identify the correct fields or the search criteria.

Question 17(d)(i): Some candidates obtained full marks in this question which was designed to be challenging. Several candidates identified the jobTitle and appRef data, but did not sort the Manager, Clerk and Administrator into descending order of jobTitle, or provide the output headings. The majority of candidates either scored 0 or 4-5 marks suggesting that the full range of marks was less accessible for C grade candidates than intended.

Question 17(d)(ii): Less than half of candidates could describe how the expected output could be used to check that the SQL statement worked correctly.

Question 18(b)(i): A few candidates could state the type of selector used.

Question 18(c): Few candidates recognised that the link used relative addressing instead of absolute addressing.

Question 18(d)(ii): The majority of candidates correctly identified the graphic file name, but did not identify the correct relative location.

Question 19(a): Less than half of candidates identified the logic operator used in the code.

Question 19(b)(i): Only half of candidates identified the type of error that occurred.

Question 19(b)(ii): Several candidates did not attempt this question. Of those who did, common errors in candidates' solutions were incorrect loop criteria, or not decrementing the balls. This was intended to be a challenging question and candidates need to read the problem carefully in order to design a correct solution.

Question 19(c)(i): Few candidates achieved marks demonstrating a limited understanding of the 'input validation' standard algorithm.

Assignment

Task 2(a): While the majority of candidates achieved one or two marks, few candidates achieved all three marks. Markers' reported that candidates sometimes repeated information given in the task rather than identifying the missing information.

Task 2(e): Candidates' ability to evaluate their own code improved slightly from last session's assignment. Candidates were unable to access marks where they made generic statements and failed to reference their own code in their responses. Very few candidates correctly identified problems with the fitness for purpose of the solution in their evaluation.

Task 3(a): Many candidates found it difficult to identify functional requirements, often writing requirements from the user's point of view instead.

Section 3: preparing candidates for future assessment

Question paper

It is evident that candidates often lack confidence, or did not appreciate the depth required in answering questions that require the application of knowledge and understanding. Candidates should be supported by using the appropriate technical vocabulary in the classroom. When designing databases, use vocabulary such as ‘the primary key of acttCode has been used because it uniquely identifies the….’.

When coding, use vocabulary such as:

- ‘we are using the logical operator AND in this selection construct.’
- ‘which loop construct does this code use and why?’
- ‘this code has been interpreted, what would happen if it had been compiled and what are the benefits?’

Centres should ensure that candidates can identify and show an understanding of the standard algorithms stated in the course specification.

Database implementation requires candidates to be able to read and explain code that makes use of the SQL stated in the course specification. In order to prepare candidates for these types of questions it would be beneficial to practice anticipating the screen output before keying in the SQL code and executing it.

In the analysis stage of database design and development and web design and development, candidates would benefit from developing a clear understanding of the difference between functional and end-user requirements. The functional requirements are what the system should be able to do, the end-user requirements are what the user can do.

Centres should ensure that candidates understand how the computer systems topic integrates into all elements of the course.

Assignment

Centres should continue to ensure that candidates are taught to implement SQL as stated in the course specification. Candidates need to be aware that a join is required where an SQL query includes data from more than one table.

It is important that candidates know to follow the design that they are required to code.

Centres should ensure that candidates are familiar with the idea of, and are able to, evaluate their own program code. Many candidates still provide generic, rote learned evaluations.

Candidates would benefit from practice in examining wireframes designs carefully and identifying where a single CSS rule can be applied multiple times to page content.

In future, candidates should ensure their requirements answers clearly refer to what the code should produce (functional), or what the user wishes to see/do (user).

Centres should ensure candidates know what evidence they need to provide for the task and how to generate it. They should encourage candidates to use the evidence checklist to ensure they gather and submit all evidence.

Grade boundary and statistical information:

Statistical information: update on courses

Number of resulted entries in 2018	6442

Number of resulted entries in 2019	6344

Statistical information: performance of candidates

Distribution of course awards including grade boundaries

Distribution of course awards	Percentage	Cumulative %	Number of candidates	Lowest mark
Maximum mark				
A	31.5%	31.5%	2000	112
B	22.4%	53.9%	1421	95
C	20.8%	74.7%	1319	78
D	14.2%	89.0%	904	61
No award	11.0%	-	700	-

General commentary on grade boundaries

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

SQA aims to set examinations and create marking instructions that allow:

- a competent candidate to score a minimum of 50% of the available marks (the notional C boundary)
- a well-prepared, very competent candidate to score at least 70% of the available marks (the notional 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 every year for each subject at each level to bring together all the information available (statistical and judgemental). The principal assessor and SQA qualifications manager meet with the relevant SQA head of service and statistician to discuss the evidence and make decisions. Members of the SQA management team chair these meetings. SQA can adjust the grade boundaries as a result of the meetings. This allows the pass rate to be unaffected in circumstances where there is evidence that the question paper has been more, or less, challenging than usual.

- The grade boundaries can be adjusted downwards if there is evidence that the question paper is more challenging than usual.
- The grade boundaries can be adjusted upwards if there is evidence that the exam is less challenging than usual.
- Where standards 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 to year. This is because the particular questions, and the mix of questions, are different. This is also the case for question papers set by centres. If SQA alters a boundary, this does not mean that centres should necessarily alter their boundary in the question papers that they set themselves.