Course report 2019

Subject	Computing Science
Level	Higher

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

Taking into account the significant differences in the way that content was assessed in session 2018-19, centres and candidates handled the question paper well. Candidates gained marks in line with expectation. This was encouraging, given the different contexts for the paper when compared with past papers.

The different nature of this paper was considered when setting grade boundaries. It is envisaged that, as candidates are able to access more question papers and centres engage with Understanding Standards events, there will be an improvement in candidate performance.

The question paper performed broadly as expected. However, statistical analysis suggests that questions 2, 14(b) and 18(c)(ii) proved to be more challenging than intended. The restructuring of course content, coupled with a lack of exemplification available to highlight assessment standards, had an impact on candidate performance. This was taken into account when setting the grade boundaries.

Assignment

The assignment performed in line with expectations. Feedback from markers, teachers and lecturers indicates it was positively received by centres, was fair and accessible for candidates. The majority of candidates understood what was required and completed the three tasks in the allocated time. The marks gained in the assignment were in line with the performance in the question paper. No change was made to grade boundaries in relation to the assignment.

Section 2: comments on candidate performance

Areas that candidates performed well in

Markers reported that candidates were well prepared. Implementation, whether in software, database or web design and development, was the strongest performing area.

Question paper

Question 4:	Most candidates gained some marks for floating-point representation. Common mistakes were either not allocating a total of 16 bits for sign and exponent, or incorrectly assigning the exponent as a negative.
Question 5:	Most candidates had a good understanding of agile methodologies, but some candidates did not pay attention to the role of the client.
Question 9:	Most candidates answered this well, but some did not name the element, responding with ‘navigation bar’ instead of NAV.
Question 11(b):	The majority of candidates could define a suitable data structure and could declare the variable based on that data structure in their chosen programming language. However, some candidates still confused the data structure with their variable name.
Question 11(c):	The majority of candidates were well prepared and could design algorithms in an unfamiliar context.
Question 12(a):	Most candidates understood how to use grouping selectors.
Question 12(b):	The majority of candidates could design an appropriate wireframe. However, many candidates did not access the fourth mark for including a ‘submit’ button.
Question 12(d)(ii):	Most candidates could identify the error in the HTML code.
Question 12(f):	Most candidates understood compatibility issues for a website.
Question 13(b):	Most candidates could draw an entity-relationship diagram for the given tables.
Question 13(c):	Most candidates used the correct aggregate function with an appropriate GROUP BY.
Question 13(e):	The majority of candidates understood the use of public and private keys in encryption.
Question 16(a):	Most candidates could design a suitable multi-level structure.
Question 18(d):	Most candidates could state the effect of a DOS attack on customers.

Assignment

Task 1(a):	The majority of candidates were able to use the information supplied to complete the entity-occurrence diagram.
Task 1(b)(i):	The majority of candidates were able to implement the SQL statement requiring an alias, a calculation and an equi-join to calculate tax for a booking.
Task 1(b)(ii):	The majority of candidates were able to implement the SQL statement(s) requiring an aggregate function, an equi-join and two queries.
Task 2(c)(i):	The majority of candidates were able to implement a modular program with appropriate procedures. The majority of candidates made correct use of a record data structure and passed appropriate parameters between sub-procedures.
Task 3(b)(ii):	The majority of candidates were able to implement Javascript functions related to mouse events to show and hide sections of a web page.

Areas that candidates found demanding

Question paper

Question 2:	The stem of this question contained some SQA Reference Language intended to support candidates. It explicitly stated that candidates should use substrings in a programming language of their choice. Many candidates were unfamiliar with the use of substrings, particularly when they attempted to answer using reference language, rather than their chosen language.
Question 7:	While many candidates appeared to have an understanding of a compound key, responses often lacked the technical accuracy to access the mark. Candidates frequently defined it as two foreign keys, instead of ‘two or more’.
Question 8(a):	The wireframe presented to candidates had several specific weaknesses as a user-interface design. The majority of candidate responses were too generic to gain any marks.
Question 10(c):	Many candidates could identify the correct return value. However, they could not recognise the logic error in returning the index of the first value in the list when the product code entered was not present in the list.
Question 11(a):	The majority of candidates restated the stem of the question instead of examining the context given to identify boundaries.

Question 12(d)(i): The majority of candidates could not implement a range check in HTML.

Question 15(a): A significant number of candidates struggled with declaring parallel arrays.

Question 16(c)(i): Most candidates had difficulty explaining the use of descendant selectors.

Question 16(e): When asked about the use of artificial intelligence in a heating system, most candidates described an aspect of a heating system that did not require intelligence.

Question 17(a): Most candidates could identify the fields including the aggregate function, but some struggled with the search criteria (date range) and the required grouping.

Question 17(c): Although many candidates were good on the syntax of the SELECT statements, they were less adept with the UPDATE command.

Question 18(a): Parameter passing was a difficult concept for most candidates, and they continued to struggle with identifying actual and formal parameters.

Question 18(c)(i): This was designed to be a challenging question. Candidates tended to trace values instead of identifying values of variables when the breakpoint occurred.

Question 18(c)(ii): Explaining code is a challenging higher-order skill. Most candidates rewrote the lines of code in sentences, paying little attention to the context of the question and its underlying operation.

Assignment

Candidate responses to tasks requiring written analysis, design and evaluation were generally weaker than responses to implementation tasks.

Task 1(c): Many candidates failed to respond within the context of the Flight table.

Task 2(b): The majority of candidates were not able to correctly indicate data flow in the design. Data flow identified in the design stage is implemented using parameter passing.

Task 2(c)(i): Many candidates failed to implement a function to return the furthest value. Candidates instead used a sub-procedure to return the value using parameter passing.

Task 2(d): Many candidates were unable to produce a trace table using variables and data structures from their own code.

Task 2(e):	The majority of candidates did not evaluate within the context of their own program code. Instead they provided rote-learned, generic evaluation definitions.
Task 3(c):	Many candidates did not provide a description of comprehensive testing of the form. Instead they offered only generic testing definitions, many of which did not extend beyond National 5 standard.

Section 3: preparing candidates for future assessment

Question paper

Centres should:

- work to improve the quality of descriptions and explanations in all areas of the course
- encourage candidates to use the correct vocabulary and technical language for the subject
- give candidates opportunities to develop their skills by presenting different coding scenarios and answering them in the context of that scenario — for example explaining the use of descendant selectors for the CSS code presented, or explaining the operation of a particular IF statement by examining the current values of variables

Centres have improved candidates' performance in the implementation of code for all topics. This is also true, to a lesser extent, for design. However, candidates need the opportunity to explore the problem solving that lies behind other stages of software development, particularly in areas such as:

- identifying inputs, outputs, scope and boundaries
- understanding data flow and its role in informing parameter passing
- testing and the use of trace tables, breakpoints and watchpoints to inform the evaluation of code
- using the context of the code in its evaluation against the various criteria — for example, candidates know the definitions of efficiency or maintainability, but should be able to explain why a particular section of code does or does not exhibit these criteria.

Centres should encourage candidates to answer in their chosen programming language. Marks are awarded for the correct constructs, regardless of programming language (including SQA Reference Language). However, it is important to note that, while questions are presented in SQA Reference Language, candidates are not expected to answer in this language. If they do, they often find it difficult to answer the question as they have not been taught the content in this language.

Centres have prepared candidates well in the use of SQL, and particularly the SELECT statement. However, centres should also encourage candidates to be alert to the use of GROUP BY, both in the design and implementation of SQL commands. The same rigour in preparing candidates should be applied to the INSERT, UPDATE and DELETE commands so that candidates are as familiar with the structure and syntax of these SQL operations.

Assignment

Most of the printed evidence submitted was of an appropriate size for markers to read. Centres should ensure that each piece of evidence includes the candidate's name. This should be in the form of comments in code, and headers or footers in word-processed evidence. Centres should encourage candidates to use the checklist provided to ensure that all evidence is submitted.

Centres should ensure that candidates are taught to implement SQL as stated in the course specification. Candidates were unable to access certain marks if they used an inner-join or used the inbuilt features of database application software to carry out the required query and then generate SQL.

Centres should remind candidates to follow the top-level design provided when implementing the software development task.

Centres should ensure that candidates appreciate the value of the analysis, design, testing and evaluation stages of the development process in the assignment. Candidates should complete these sections within the context of the development task.

Grade boundary and statistical information:

Statistical information: update on courses

Number of resulted entries in 2018	4099

Number of resulted entries in 2019	3228

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	23.2%	23.2%	748	108
B	20.8%	44.0%	672	91
C	20.0%	63.9%	644	74
D	17.3%	81.3%	559	57
No award	18.7%	-	605	-

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.