Course Report 2018

Subject	Computing Science
Level	National 5

This report provides information on the performance of candidates. 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

Summary of the course assessment

Component 1: question paper

The question paper had two sections and was worth 110 marks. It incorporated a mixture of short response and extended response questions, reflecting the specimen question paper.

The question paper largely performed in line with expectations and feedback from the markers, teachers and lecturers suggested that it was fair in terms of course coverage. However, analysis of the statistical data indicated that the question paper was less demanding than intended for A grade candidates. This was taken into account when setting the grade boundaries.

Component 2: assignment

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

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

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

The assignment was worth 50 marks — 25 marks for ‘software design and development’ and the remaining 25 marks split between ‘database design and development’ and ‘web design and development’.

The assignment largely performed as expected, with candidates achieving the anticipated range of marks on most tasks.

The data dictionary task 1(b) did not differentiate effectively, as most candidates achieved full marks. The software evaluation task 2(d) was more challenging than expected for candidates. This was taken into account when setting the grade boundaries.

Section 2: comments on candidate performance

Areas in which candidates performed well

Component 1: question paper

Section 1:

Questions 1 and 3: Most candidates knew file types and their use.

Question 2: Candidates had to state two reasons why the web page was not 'fit for purpose'. In general, candidates answered this question well, although some responses were about consistency rather than fitness for purpose.

Question 4(a): Most candidates correctly read the code.

Question 6: Most candidates understood the use of encryption.

Question 8: Most candidates were clear on consistency of websites.

Question 9: Most candidates demonstrated they understood both JavaScript and onmouseover.

Question 10(a): Most candidates identified that any item, more specifically the DVD, and not just the books would be changed.

Question 10(b): Most candidates correctly rewrote the SQL statement, however, some did not identify that 'book' is a string. This question is similar to question 17 in the specimen question paper.

Question 11: Most candidates correctly identified the condition in the design.

Question 13: Most candidates knew and understood the Copyright, Designs and Patents Act.

Question 14(a): Most candidates provided good explanations and demonstrated they could read and understand code.

Question 14(c): Most candidates knew how to read this code.

Section 2:

Question 17 (a)(i): Candidates referred to either the 1D array or the Boolean. Although they demonstrated they knew and understood the part they responded to, candidates need to refer to both if the question has 2 marks.

Question 17(a)(ii): Most candidates provided a code that created a total, but several candidates did not take into account 'a running total within a loop'. This is one of the standard algorithms in the course.

Question 18(a): Most candidates produced a clear hierarchical structure.

Question 18(b):	Most responses showed candidates clearly understood the elements within HTML.
Question 18(c):	Most responses demonstrated candidates understood jpeg files; however, some candidates gave the same reason twice.
Question 18(e):	Most candidates provided the correct response. However, some candidates lacked the accuracy required, despite the scaffolding provided within the question.
Question 18(f)(i):	Almost all candidates gained a mark for this question.
Question 18(g):	Most candidates knew and understood how to test a web page.
Question 19(a):	Most candidates correctly identified the inputs, processes and outputs.
Question 19(c)(i):	Most candidates provided the correct pre-defined function.
Question 19(d):	Most candidates came up with a solution to check the password was 8 characters in length. However, some responses assumed that the user would enter the correct number of characters the second time and did not put the input request into a conditional loop.
Question 19(e):	Most candidates demonstrated they knew about testing data.
Question 20(a):	Most candidates could identify primary and foreign keys. (However, see question 5 in the 'Areas which candidates found demanding' section, where candidates did not know why they are used.)
Question 20(b):	Candidates demonstrated they could recognise the types of data used in the database.
Question 20(c):	Most candidates identified the fields and tables, but some struggled with the search criteria.
Question 20(d)(i):	Most candidates gave two of the expected outputs, but missed the third.
Question 21(b)(i):	Most candidates correctly stated the outputs.
Question 21(b)(ii):	Most candidates could explain how to apply coding skills to recognise incorrect data input.
Question 21(b)(iii):	Most candidates correctly stated the component that calculates the total cost.
Question 21(b)(iv):	Most candidates correctly named the part of the computer system.
Question 22(a)(i):	Most candidates knew how to convert to an 8-bit binary number.

Question 22(b)((i): Most candidates were clear in identifying the object.

Question 22(b)(ii): Most candidates could identify the attributes.

Question 22(c): Most candidates knew how to identify features and functions of computer systems.

Question 23(a): Most candidates answered this question well. The question listed what candidates had to do; however, many did not name the relationship and several did not identify the primary and foreign keys.

Question 23(b)(ii): Most candidates understood validation.

Question 23(c)(ii): Most candidates managed to position the top section with the required text. However, in the second section, some candidates did not number the list or identify any links.

Component 2: assignment

Task 1(a): Most candidates performed very well in this task, gaining full marks. Markers' feedback stated that candidates lost marks because they did not identify that the job information should also be included as inputs.

Task 1(b): Almost all candidates have a very good understanding of how to complete a data dictionary.

Task 1(c) Most candidates clearly demonstrated that they could create a database table and assign validation to fields. Markers feedback stated that candidates lost marks because they provided incomplete evidence.

Task 1(d): Most candidates were capable of writing and implementing an SQL query. However, more candidates did not submit any evidence for this task than for any other task.

Task 2(b): Most candidates performed well in this task, demonstrating a good understanding of test data within the context of the problem.

Task 3(b): Most candidates demonstrated an excellent ability to write and test both HTML and CSS code. Some candidates did not gain marks for implementing an external CSS file, as they had implemented internal CSS code instead.

Areas which candidates found demanding

Component 1: question paper

Although candidates could implement a practical solution in the assignment, the question paper showed that many candidates were unable to relate to the theory behind what they were doing. For example, candidates performed very well in the data dictionary task in the assignment, but in the question paper, candidates were unsure why they created a data dictionary. If candidates can understand what is behind the practical tasks, this will help them respond to the question paper.

Section 1:

Question 4(b): Although most candidates managed to gain marks for this question, some missed out the mantissa and others missed out the exponent. This type of question was also in the specimen question paper.

Question 5: Most candidates demonstrated they knew what a primary key was in both the assignment and in question 20(a). However, many candidates' explanations of the need for a primary key were unclear and they often explained the foreign key instead.

Question 7: Most candidates were clear on the pre-defined function, but could not identify the parameter(s) required.

Question 12: Although candidates have used low-fidelity in the assignment and there are examples of low-fidelity later in the paper, many candidates did not demonstrate they understood why they are used.

Question 14(b): Although candidates use and apply logical operators in coding and SQL, many could not identify the logical operators in the question.

Question 15: Many candidates knew what a conditional loop was, but did not know why you would use a loop when writing code.

Question 16: Many candidates identified that the SQL statement would delete the competitor when it was run, but not that it would delete the whole row. When answering questions that apply knowledge of how code performs, candidates must be technically accurate.

Section 2:

Question 17(b): Many candidates still struggled with translation. They were not clear why the program would use compiled code. This style of question was also in the specimen question paper.

Question 18(d): This was one of the more challenging questions, testing the A grade criteria. Many candidates had difficulty applying their knowledge of CSS and HTML, including how they work together and how they are used.

Question 18(f)(ii): Many candidates did not know the types of addressing used in hyperlinks.

Question 19(b): Many candidates were not clear how to create a user interface design.

Question 19(c)(ii): Many candidates had difficulty explaining why you would use a pre-defined function.

Question 20(d)(ii): Some candidates struggled to provide a sufficient description.

Question 20(e): Although candidates applied referential integrity in the assignment, some struggled to describe a problem that could occur when it is not used.

Question 21(a): Candidates used a range of design methodologies to produce a solution, although some refined other steps rather than just step 4, as asked for in the question.

Most candidates achieved around half of the available marks. Typically, candidates gained no marks where they did not apply the appropriate conditions to the problem. For example '2 or more adults should have given (>=), more than 2 children should have given (>2)'.

Question 23(b)(i): Although candidates used a data dictionary in the assignment and gave responses on databases elsewhere in the paper, the majority could not state the purpose of a data dictionary.

Question 23(c)(i): Most candidates did not clearly explain the purpose of the section of code.

Component 2: assignment

Task 2(a): Many candidates performed well in this task. However, some found implementing the supplied structure diagram challenging and chose to devise their own solutions instead.

Task 2(c): Candidates found this task challenging and few gained full marks. They often identified the outer extremes of the test data (0, 100), but did not identify the extremes for each individual signal strength.

Task 2(d): Candidates found this task the most challenging. Many candidates:

- did not evaluate their own code, instead making generic statements regarding evaluation.
- made errors in their code that were not acknowledged when evaluating the fitness for purpose of their code.
- stated that they had implemented part of the problem, although their code clearly showed they had not. For example, candidates stating that their program was robust because they had used input validation to ensure the user could only enter valid data. However, on examination, the code showed they had not implemented input validation.

Task 3(a): Although candidates' responses indicated that they had read the task, they struggled to identify the difference between end-user and functional requirements. Candidates often provided single-word responses.

Section 3: advice for the preparation of future candidates

Component 1: question paper

Overall, candidates performed well in the question paper, with many demonstrating they understood the full course content. Centres should continue to offer support and preparation across all content, as this helps candidates achieve high marks in this part of the course assessment.

A key message is that not all candidates understand or can explain why they carry out practical work. For example, candidates showed they can use a data dictionary in the assignment, but did not appear to know why they were using it or its actual purpose.

When focussing on practical tasks in the classroom, centres are encouraged to ask candidates to think about what the code actually does when executed. For example:

- What is the effect of HTML and CSS?
- What is the effect of certain coding constructs and the standard algorithms?
- What is the effect of SQL in a database environment?

This should help to embed understanding of the practical elements of the courses.

Component 2: assignment

Centres should ensure candidates know what evidence they need to provide for the task and how to generate it. Encourage candidates to use the evidence checklist to ensure they gather and submit all evidence. Candidates who used the checklist tended to have more complete evidence than those who did not.

In particular, candidates must provide complete evidence for the database implementation task. Many candidates focussed on evidence of validation (range check and restricted choice), but often did not provide evidence of field types and sizes.

Centres should ensure that candidates are taught to implement SQL, where required. Candidates gained no marks if they used the inbuilt features of database application software to carry out the required query and then generate SQL.

Candidates occasionally supplied evidence of program testing that did not match the test data that they had created. Candidates should be aware that they must supply evidence of their test data.

Centres should prepare candidates to evaluate their own program code. Candidates should avoid making generic statements regarding fitness for purpose, efficiency, robustness and readability.

Candidates must clearly differentiate between end-user and functional requirements. Markers feedback stated that some candidates provided similar responses for both.

Centres should prepare candidates to create either internal or external CSS code. Some candidates gained no marks, as they had not implemented an external CSS file, as required for the task. Future tasks may ask for internal CSS code.

If candidates supply screenshots of their work as evidence (especially code), they should ensure their evidence is clear and legible. Some candidates had shrunk screenshot graphics to a point where the text became unreadable.

Grade boundary and statistical information:

Statistical information: update on courses

Number of resulted entries in 2017	7442

Number of resulted entries in 2018	6442

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	29.2%	29.2%	1884	114
B	24.2%	53.4%	1559	97
C	21.3%	74.7%	1371	80
D	14.3%	89.0%	919	63
No award	11.0%	-	709	-

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 which allow a competent candidate to score a minimum of 50% of the available marks (the notional C boundary) and 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 Business Manager and Statistician to discuss the evidence and make decisions. The meetings are chaired by members of the management team at SQA.

- The grade boundaries can be adjusted downwards if there is evidence that the exam is more challenging than usual, allowing the pass rate to be unaffected by this circumstance.
- The grade boundaries can be adjusted upwards if there is evidence that the exam is less challenging than usual, allowing the pass rate to be unaffected by this circumstance.
- Where standards are comparable to previous years, similar grade boundaries are maintained.

Grade boundaries from exam 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 exams set by centres. If SQA alters a boundary, this does not mean that centres should necessarily alter their boundary in the corresponding practice exam paper.