Course report 2022

Subject	Computing Science
Level	Advanced 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 appeals.

Grade boundary and statistical information

Statistical information: update on courses

Number of resulted entries in 2022	695

Statistical information: performance of candidates

Distribution of course awards including grade boundaries

A	Percentage	36.7	Cumulative percentage	36.7	Number of candidates	255	Minimum mark required	94
B	Percentage	21.3	Cumulative percentage	58.0	Number of candidates	150	Minimum mark required	79
C	Percentage	18.4	Cumulative percentage	76.4	Number of candidates	125	Minimum mark required	65
D	Percentage	14.8	Cumulative percentage	91.2	Number of candidates	105	Minimum mark required	50
No award	Percentage	8.8	Cumulative percentage	N/A	Number of candidates	60	Minimum mark required	N/A

You can read the general commentary on grade boundaries in appendix 1 of this report.

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 more statistical reports on the statistics page of SQA’s website.

Section 1: comments on the assessment

Question paper

The 2021–22 question paper provided candidates with an option to answer either the 'Database design and development' section or the 'Web design and development' section. The 'Database design and development' section was answered by 49% of candidates and the 'Web design and development' section was answered by 51% of candidates. The average mark for each section was very similar.

Statistical evidence suggests that the introduction of grade C level questions at the start of each section helped candidates to focus on the specialist content of each section, before tackling the more demanding problem-solving questions that followed.

Most questions performed as expected, however some were more demanding than intended and grade boundaries were adjusted accordingly.

Project

The project performed as intended and there were no adjustments made to the grade boundary.

Section 2: comments on candidate performance

Areas that candidates performed well in

Question paper

Software design and development

Question 2(b): Most candidates were able to state that the array had not been sorted.

Question 3: Most candidates were able to write down the result array after the second iteration of the insertion sort, and after the second pass of the bubble sort.

Question 4(a), (b)(i), (c), (d)(ii): Most candidates demonstrated a good knowledge of the object-oriented programming concepts of inheritance, constructor methods, polymorphism, and arrays of objects in their responses to this question.

Question 4(d)(iii): Many candidates were able to achieve marks for answers that were partially correct.

Question 5(a)(iii), (b), (c): Most candidates were able to achieve marks for answers that were partially correct and demonstrated good knowledge and understanding of the need to use nested loops when working with 2D arrays.

Database design and development

Question 6: Most candidates were able to achieve 1 mark for stating appropriate data types and lengths for firstName, lastName and contractType fields.

Question 8(c): Most candidates identified the use made of the BETWEEN operator, and some candidates correctly identified the use made of HAVING COUNT(*)>=3.

Web design and development

Question 11(b): Most candidates demonstrated good knowledge of matching the HTML form method to the PHP $GET() function.

Question 11(c): Many candidates demonstrated their knowledge of how the mysqli_connect() function is used to establish a connection with a database server. However, some candidates failed to list the parameters of the function in the correct order.

Areas that candidates found demanding

Question paper

Software design and development

Question 1: Although most candidates were able to identify the data structure as being a linked list, many candidates failed to state that it was a **double** linked list.

Question 4(b)(ii): Many candidates were able to demonstrate their knowledge of inheritance in their responses, but some candidates failed to indicate the intended data types for the additional instance variables of hoursInService and hoursToNextService.

Question 4(d)(i): Although many candidates were able to state that the code was used to instantiate a new object belonging to the Plane class, only a few candidates matched the listed values to the relevant instance variables.

Question 5(a)(i): Many candidates failed to identify functional requirements that were additional to those already stated in the question stem, or incorrectly suggested end-user requirements rather than the technical functional requirements that were expected.

Question 5(a)(ii): Although candidates were able to state the correct dimensions of a suitable 2D array, many failed to indicate the intended data type.

Database design and development

Question 7: Most candidates failed to demonstrate any understanding of a surrogate key and the possible benefits its introduction would bring.

Question 8(a)(i): Many candidates failed to identify the correct type of feasibility and support their answer with appropriate justification.

Question 8(b): Very few candidates achieved full marks for their ERDs. Identification of mandatory and optional participation was particularly poor.

Question 8(d)(i): Only some candidates were able to identify the need to use the ANY operator in the missing search criteria.

Question 8(f)(i): Although a few candidates were able to describe the use that would be made of the persona and test case during testing, most were unable to identify that this would take place during final testing.

Question 8(f)(ii): Many candidates failed to refer to the screenshot evidence provided in the question stem in their explanations of whether the solution was fit for purpose.

Web design and development

Question 10: Although many candidates did receive partial marks for this question, only some candidates demonstrated knowledge of how session variables are used to share values across multiple pages of a website, and referred to a login system in their responses.

Question 11(a)(i): Many candidates failed to identify the correct type of feasibility and support their answer with appropriate justification.

Question 11(d): Although most candidates did receive marks for partially correct answers, many candidates failed to identify the use made of the PHP mysqli_query() and mysqli_num_rows() functions.

Question 11(e): Although many candidates did receive partial marks for their descriptions, some candidates failed to mention the use that would be made of the CSS @media print rule, rather than @media screen, to achieve the desired paper output.

Question 11(f)(i): Although a few candidates were able to describe the use that would be made of the persona and test case during testing, most were unable to identify that this would take place during final testing.

Question 11(f)(ii): Many candidates failed to refer to the screenshot evidence provided in the stem in their explanations of whether the solution was fit for purpose.

Areas that candidates performed well in or found demanding

Project

Stage 1: Analysis

Overall, candidates did well in this stage of their project work. Scope, boundaries and constraints were completed well. In addition, many UML case diagrams helped candidates to identify the end-user and backend functional requirements of their solutions. Most candidates produced a project plan that identified tasks to be completed at each stage of the developments, together with an estimate of the timings of each task.

Description of the problem

Many candidates failed to provide the details necessary in the outline of their problem. In this part of their solution, candidates must clearly identify **all** of the Advanced Higher concepts and integration that they intend to include in their solution. Their outlines should refer to each of the relevant Advanced Higher competences and skills listed in the mandatory requirements diagrams in the coursework assessment task.

Requirements specification

Although most candidates received good marks for this part of their analysis, many failed to include input validation as a functional requirement.

Stage 2: Design

In general, candidates who intended to integrate with a database or web page, completed the necessary design work well. Most candidates' user-interface design work accurately matched their requirements listed at the 'Analysis' stage.

'Design of Advanced Higher concepts' and 'Overall design matches the requirements specification'

Many candidates completing software design and development projects simply copied, pasted and numbered their code and then presented this as pseudocode. Where this was the case, it was not possible to award any marks for the design of Advanced Higher algorithms or for the design of any other requirements listed in the requirements specification. Few candidates who planned to implement a procedural program produced a top-level design to indicate the data flow between modules.

User-interface design

Although most candidates received marks for partially complete user-interface designs, some failed to indicate underlying processes that would be used to receive and process input received or generate the output displayed. Having omitted essential input validation from their requirements, some candidates also failed to indicate the intended validation, and how users would be alerted to input errors. This resulted in a consequential error that was not penalised again in later stages; however, it is important evidence that candidates are expected to indicate in their designs.

Stage 3: Implementation

It was very clear that most candidates enjoyed coding their solutions and they achieved good marks for implementation of the Advanced Higher concepts, the integration and user-interface.

Description of new skills and/or knowledge researched and developed

Some candidates did not refer to skills that go beyond what is required for the Advanced Higher course. Instead, they referred to course requirements such as integrating with another area of the course, learning PHP, or even skills from Higher or National 5 level. Similarly, few candidates referred to the need to learn additional frameworks, plug-ins or even new programming languages.

Log of ongoing testing

Some candidates failed to provide sufficient evidence of testing that was carried out as the solution was being implemented.

Stage 4: Testing

Most candidates gained good marks for test plans that indicated all the requirements listed in the 'Analysis' stage and descriptions of the tests planned to demonstrate that each requirement was thoroughly tested. In addition, most candidates generated the evidence needed to demonstrate that all tests in their test plans had been carried out.

Persona and test cases
Although most candidates produced good test plans, some failed to provide a suitable description of a typical end-user in the form of a persona. In addition, some candidates did not provide test cases in the form of tasks to be completed by typical end-users of the finished system.

Description of results of the test cases
Rather than referring to planned test cases carried out by adopting the persona described earlier, some candidates' descriptions referred to the results of testing in general.

Stage 5: Evaluation

Many candidates received marks for descriptions that accurately referred to each of the requirements identified at the 'Analysis' stage, and also made reference to testing of the requirements when carrying out their test plan.

Maintainability and robustness
Many candidates' descriptions of the maintainability of their solution were appropriate for Higher and National 5 level, but not for Advanced Higher level, because they failed to make reference any future maintenance that may be carried out. In addition, many candidates incorrectly stated that their solutions were robust when there was no evidence of testing to support such claims.

Section 3: preparing candidates for future assessment

Question paper

Centres should ensure that candidates are able to state the correct data types required to store data. In 'Software design and development', candidates will be required to do this in a programming language of their choice. Responses in languages that make use of dynamic typing, such as LiveCode, JavaScript, PHP and Python, are acceptable, but candidates who use these languages should be encouraged to add commentary to indicate the data type that is intended.

Centres should ensure that candidates are familiar with all the PHP mysqli functions required at Advanced Higher level.

Centres should ensure that candidates have knowledge of the content of the Advanced Higher course in the areas of analysis, testing and evaluation. Although there is a requirement to demonstrate this knowledge in their project work, some content from these areas of the course will always be sampled in the question paper. For example:

- In question 5(a)(i), many candidates suggested end-user requirements rather than functional requirements that related to technical aspects of the solution, such as back-end processing, that would be required in the solution.
- In questions 8(a)(i) and 11(a)(i), candidates were expected to identify the feasibility study that had been highlighted in the stem for part (a). In their responses, it was clear that some candidates were unfamiliar with the types of feasibility listed in the Advanced Higher Computing Science Course Specification.
- In questions 8(f)(i) and 11(f)(i), most candidates failed to demonstrate an understanding of how personas and test cases are used by the development team in their final testing of any solution.

Overall, candidates coped well with all problem-solving questions with challenging, unseen programming tasks, such as questions 5(a)(iii), 5(b) and 5(c). Centres should continue to encourage candidates to attempt these more challenging questions as statistical evidence shows that most candidates received some marks, even when their overall solution was incorrect or incomplete.

Candidates' knowledge of object-oriented programming concepts continues to improve, and candidates coped well with the problem-solving required in question 4(d)(iii). Centres should continue to ensure that candidates are familiar with the required 'OO' terminology used to explain the operation and effect of code that is written in the SQA Reference Language. They should continue to ensure that candidates have opportunities to solve problems by applying standard algorithms to arrays of objects.

Project

Coursework assessment task

Centres must ensure that the correct version of the coursework assessment task is being used. This gives good advice to candidates about what evidence they are expected to submit at each stage of the project. This has been updated for session 2022–23 to incorporate some of the messages in this report into the guidance for teachers and lecturers and guidance for candidates.

In general, candidates who followed this advice closely received good marks for all areas of their work. However, it was clear that a significant number of centres had also referred candidates to guidelines issued for the previous version of the project. This was to the detriment of candidates, who spent a lot of time completing work and generating evidence, such as feasibility studies and end-user surveys, that was completely unnecessary and received no marks.

Project selection

All candidates should be supported when selecting a project to complete. This helps to ensure that the chosen development meets the criteria for an Advanced Higher project, and is achievable in the time available. Where a project does not meet Advanced Higher criteria, a number of marks in every stage of the project are not accessible to candidates.

As well as meeting the criteria, it is helpful for candidates to have a clear idea of what project combination they are completing. The mandatory requirements diagrams in the coursework assessment task should be used as a guide.

Presentation of evidence

Prior to submission, it would be extremely helpful if centres could ensure that candidates present their evidence in clearly labelled sections arranged in the same sequence as the marking instructions. This helps with marking but also helps candidates to identify any omissions.

To award marks for ‘Implementation’, markers **must** be able to read all the code that has been independently generated by a candidate. Screenshots of code that are reduced to fit an A4 printed page are almost impossible to read, especially those screenshots of white or coloured code on a black background. Rather than providing screenshots, if it is not possible to print code directly from within the development environment used, centres should advise candidates to copy their code into word processing software and print it from there. When the code has been printed and is easily read, the candidate can then annotate or highlight their code to indicate where in the solution the essential Advanced Higher techniques have been applied, and where integration with at least one other area of the course content has been achieved.

Automatically generated code

Many projects submitted relied heavily on frameworks and/or software plug-ins. The code generated automatically by such software solutions often ran to over 100 pages. Where this code was unannotated, it was unclear which parts, if any, of the solution had been generated by the candidates themselves. As a result, it was very difficult for markers to award marks for

the Advanced Higher techniques and integration that are necessary for any Advanced Higher project. Where code runs to a significant number of pages, centres should encourage candidates to highlight and annotate the code that relates to the Advanced Higher concepts, integration and other functional requirements. This code could also be extracted, with the full code provided as an appendix. This will help marking but will also help candidates to focus on the important aspects of the projects.

Requirements specification

When creating their requirements specification, all candidates should include input validation as a functional requirement. This is an essential requirement of all Advanced Higher projects. If a candidate wishes to implement a solution that relies on keypresses, then these keypresses must be validated, and appropriate actions must be taken to alert users to the use of inappropriate keypresses. In addition, candidates should be encouraged to give more thought to the **functional requirements** of the project to ensure that the essential Advanced Higher competences and skills are present in the final solution. Although end-user requirements are important, the project does not require any end-user surveys to be completed, or any end-user testing to be carried out.

Good practice noted by markers includes the numbering of end-user and functional requirements so that they could easily be cross-referenced in future stages of the development work. This is especially relevant when creating the test plan as part of the evidence submitted for the 'Testing' stage.

Test plan

Centres should encourage their candidates to design their test plan **before** starting their implementation work, so they can consider what needs to be tested as part of final testing before they became distracted by any problems or issues that they encounter during the implementation stage.

Evidence required for implementation

Any candidate who intends to implement a database as part of their solution must include evidence that the implemented database structure matches all aspects of the design shown in the data dictionary. In addition, by capturing evidence of the initial values stored in database tables, candidates should be able to demonstrate that all implemented queries function as expected by capturing evidence of updated tables or the results of searches. Similarly, candidates who implement an Advanced Higher algorithm as part of a software development project should generate evidence to demonstrate that the search and/or sort implemented functions as expected. This evidence can be included along with evidence that planned testing has been carried out.

Description of new skills and/or knowledge researched and developed

These marks are designed to reward candidates who stretch themselves with a project that will have some functional requirements that require them to research and implement something beyond the Advanced Higher course. This is intended to be tied to functionality and not something trivial and cosmetic. Some candidates are better advised not to be distracted with this, and to focus on the Advanced Higher concepts, integration and user-interface, where most of the marks are available.

Ongoing testing

Centres should encourage candidates to test their solution as they develop it. As new modules, processes, web pages or database components are added to a solution, candidates are likely to check that these new components are functioning correctly. By capturing evidence of these checks, candidates are generating evidence of ongoing testing that should be presented along with other implementation evidence.

In the section of the project for the results of test cases, marks are awarded for descriptions that indicate that candidates understand how a persona and test cases are used when final testing is carried out. These descriptions must refer to the test cases that were carried out when adopting that persona described in the test plan.

Evaluation

When evaluating whether their solutions are fit for purpose, candidates must refer to the requirements identified at the 'Analysis' stage. Simply stating that all requirements have been met is not sufficient to gain marks at Advanced Higher level. Instead, candidates are expected to refer to individual requirements and discuss the extent to which they believe that they have been met. When discussing the maintainability of their solutions, candidates must refer to different types of maintenance that may be carried out in future, and explain how features of their solution (such as commentary, indentation and modularity) will support and aid that future maintenance. In their evaluations of robustness, candidates are expected to refer to the input validation that was implemented and testing of the validation that was carried out.

Appendix 1: 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.

Grade boundaries from question papers in the same subject at the same level tend to be marginally different year on year. This is because the specific questions, and the mix of questions, are different and this has an impact on candidate performance.

This year, a package of support measures including assessment modifications and revision support, was introduced to support candidates as they returned to formal national exams and other forms of external assessment. This was designed to address the ongoing disruption to learning and teaching that young people have experienced as a result of the COVID-19 pandemic. In addition, SQA adopted a more generous approach to grading for National 5, Higher and Advanced Higher courses than it would do in a normal exam year, to help ensure fairness for candidates while maintaining standards. This is in recognition of the fact that those preparing for and sitting exams have done so in very different circumstances from those who sat exams in 2019.

The key difference this year is that decisions about where the grade boundaries have been set have also been influenced, where necessary and where appropriate, by the unique circumstances in 2022. On a course-by-course basis, SQA has determined grade boundaries in a way that is fair to candidates, taking into account how the assessment (exams and coursework) has functioned and the impact of assessment modifications and revision support.

The grade boundaries used in 2022 relate to the specific experience of this year's cohort and should not be used by centres if these assessments are used in the future for exam preparation.

For full details of the approach please refer to the National Qualifications 2022 Awarding — Methodology Report.

