Advanced Higher
Course Assessment
Specification

Advanced Higher Computing Science Course Assessment Specification (C716 77)

Valid from August 2015

This edition: April 2016, version 2.1

Please refer to the note of changes at the end of this Course Assessment Specification for details of changes from previous version (where applicable).

Course outline

Course title:	Advanced Higher Computing Science
SCQF level:	7 (32 SCQF credit points)
Course code:	C716 77
Course assessment code:	X716 77

The purpose of the Course Assessment Specification is to ensure consistent and transparent assessment year on year. It describes the structure of the Course assessment and the mandatory skills, knowledge and understanding that will be assessed.

Course assessment structure

Component 1 — project	90 marks
Component 2 — question paper	60 marks
Total marks	**150 marks**

This Course includes eight SCQF credit points to allow additional time for preparation for Course assessment. The Course assessment covers the added value of the Course.

Equality and inclusion

This Course Assessment Specification has been designed to ensure that there are no unnecessary barriers to assessment. Assessments have been designed to promote equal opportunities while maintaining the integrity of the qualification.

For guidance on assessment arrangements for disabled learners and/or those with additional support needs, please follow the link to the Assessment Arrangements web page: www.sqa.org.uk/sqa/14977.html.

Guidance on inclusive approaches to delivery and assessment of this Course is provided in the *Course/Unit Support Notes*.

Assessment

To gain the award of the Course, the learner must pass all of the Units as well as the Course assessment. Course assessment will provide the basis for grading attainment in the Course award.

Course assessment

SQA will produce and give instructions for the production and conduct of Course assessments based on the information provided in this document.

Added value

The purpose of the Course assessment is to assess added value of the Course as well as confirming attainment in the Course and providing a grade. The added value for the Course will address the key purposes and aims of the Course, as defined in the Course Rationale. It will do this by addressing one or more of breadth, challenge, or application.

In this Course assessment, added value will focus on the following:

- breadth — drawing on knowledge and skills from across the Course
- challenge — requiring greater depth or extension of knowledge and/or skills
- application — requiring application of knowledge and/or skills in practical or theoretical contexts as appropriate

Through the Computing Science Units, learners will develop practical computing science skills and knowledge and understanding of key concepts and processes relating to software and information system design and development in a variety of contexts.

The added value consists of the following.

To achieve success in the Course, learners must show that they can **apply** the knowledge and skills developed through the Units, in both practical and theoretical contexts. Added value will be assessed through a project and a question paper.

The project requires learners to demonstrate aspects of challenge and application in a practical context. Learners will **apply** knowledge and skills from across the Course to plan, design, implement, evaluate and report on a solution to solve an appropriately challenging practical computing science problem.

The question paper requires learners to demonstrate aspects of breadth and application in theoretical contexts. Learners will apply breadth of knowledge from across the Course, and depth of understanding, to answer appropriately challenging questions in computing science contexts.

Grading

Course assessment will provide the basis for grading attainment in the Course award.

The Course assessment is graded A–D. The grade is determined on the basis of the total mark for all Course assessments together.

A learner’s overall grade will be determined by their performance across the Course assessment.

Grade description for C

For the award of Grade C, learners will have demonstrated successful performance in all of the Units of the Course. In the Course assessment, learners will typically have demonstrated successful performance in relation to the mandatory skills, knowledge and understanding for the Course.

Grade description for A

For the award of Grade A, learners will have demonstrated successful performance in all of the Units of the Course. In the Course assessment, learners will typically have demonstrated a consistently high level of performance in relation to the mandatory skills, knowledge and understanding for the Course.

Credit

To take account of the extended range of learning and teaching approaches, remediation, consolidation of learning and integration needed for preparation for external assessment, six SCQF credit points are available in Courses at National 5 and Higher, and eight SCQF credit points in Courses at Advanced Higher. These points will be awarded when a Grade D or better is achieved.

Structure and coverage of the Course assessment

The Course assessment will consist of two Components: a project and a question paper.

Component 1 — project

The purpose of the project is to assess practical application of knowledge and skills from across the Course to develop a solution to an appropriately challenging and complex computing science problem. It will assess learners' skills in planning and designing a solution to a problem, implementing and testing a solution, and evaluating and reporting on that solution.

The assignment will have 90 marks (60% of the total mark).

The project will be based on a meaningful task providing appropriate challenge and requiring application.

The project should clearly demonstrate application of significant knowledge and skills, at an appropriate level, from across the Course (as defined in the 'Further mandatory information on Course coverage' section of this document).

Marks will be awarded for:

- producing a detailed project plan
- producing a requirements specification and test plan
- iterative design and development of a solution
- final testing of the solution
- evaluating the solution and development process
- presenting/communicating the solution

Evidence produced by the learner should include:

- a detailed project plan, requirements specification and test plan
- a record of progress through the project, including reflective commentary and all items of evidence specified within the task (including program listings, screenshots, web page source files, data files or similar as appropriate)
- evidence of final testing
- a qualitative evaluation of the solution and development process
- a presentation or report on the solution, including possible implications
- detailed assessor observation notes, providing evidence of candidate's degree of independence and safe working

The presentation or communication of the solution can be in any suitable format: visual, oral, manual or electronic.

Component 2 — question paper
The purpose of the question paper is to assess breadth of knowledge from across the Course, depth of understanding, and application of this knowledge and understanding to answer appropriately challenging questions.

This question paper will give learners an opportunity to demonstrate the following skills, knowledge and understanding:

- the ability to apply computational thinking to understand problems across a range of contexts
- analysing complex problems within computing science across a range of contemporary contexts
- the ability to communicate how a well-structured, complex modular program works
- the ability to communicate complex computing science concepts clearly and concisely, using appropriate terminology
- knowledge and understanding of key aspects of contemporary project planning and management
- knowledge and understanding of the wide role and impact of contemporary computing technologies on society, including legal, ethical, environmental and economic implications
- knowledge and understanding of contemporary programming paradigms

The question paper will have 60 marks (40% of the total mark) and consist of structured and extended response questions.

It will give learners an opportunity to demonstrate application of knowledge and understanding to answer appropriately challenging context-based questions, by drawing on and applying knowledge from the table provided in the ‘Further mandatory information on Course coverage’ section at the end of this Course Assessment Specification.

Questions will have balanced sampling across the topics of the Course, and:

- There will be four structured questions of 15 marks each, which will draw on knowledge and understanding from two or more topics; each question will be broken down into sub-questions. Some sub-questions will require integration.
- Each of these structured questions will be set in some meaningful context and will require learners to provide some descriptions and explanations, and may involve some calculations.

Questions assessing understanding and application of programming skills will (mainly) be presented using SQA standardised reference language, which may include the following terms:

Variable types:	INTEGER, REAL, BOOLEAN, CHARACTER
Structured types:	ARRAY, STRING, RECORD
Objects:	CLASS .. IS { .. } METHODS ... END CLASS
	CLASS .. INHERITS .. WITH { ..} METHODS .. END CLASS
	CONSTRUCTOR…END CONSTRUCTOR…OVERRIDE
System entities:	DISPLAY, KEYBOARD

Variable introduction:	DECLARE…INITIALLY
Assignment:	SET … TO …
Conditions:	IF .. THEN .. (ELSE) ... END IF
Conditional repetition:	WHILE … DO … END WHILE REPEAT … UNTIL …
Fixed repetition:	REPEAT … TIMES … END REPEAT
Iteration:	FOR .. FROM .. TO .. DO .. END FOR FOR EACH … FROM … DO … END FOR EACH
Input / output: (including files)	RECEIVE … FROM … SEND … TO ..
File Operations:	OPEN… CLOSE… CREATE
Operations:	-, +, *, /, ^, mod, &
Comparisons:	=, ≠, <, ≤,, >, ≥
Logical operators:	AND, OR, NOT
Subprograms:	id(parameters)

Where required, subprograms may be presented in the following formats:

```
PROCEDURE id (parameters)
        <commands>
END PROCEDURE
```

```
FUNCTION id(parameters) RETURNS type
        <commands>
        RETURN expression
END FUNCTION
```

<….> is used to indicate code fragments which are not expressed in the reference language

# is used to indicate comments

Where learners are required to answer by writing code, answers may be expressed using any form of pseudocode, any other design notation or any programming language; marks will be awarded for demonstrating understanding, not for correctness of syntax.

For more information about the structure and coverage of this Component of the Course assessment, please refer to the Question Paper Brief.

Setting, conducting and marking of assessment

Question paper

This question paper will be set and marked by SQA, and conducted in centres under conditions specified for external examinations by SQA. Learners will complete this in 2 hours.

Controlled assessment — project

The project is:

- set by centres within SQA guidelines
- conducted under some supervision and control

Evidence will be internally marked by centre staff in line with SQA Marking Instructions.

All marking will be quality assured by SQA.

Setting the assessment

Set by centres within SQA guidelines.

The specification for the project will be generated by the learner and agreed between the learner and the teacher/lecturer.

The project will be a meaningful and appropriately challenging task, which should clearly demonstrate application of significant knowledge and skills, at an appropriate level, from across the Course (as defined in the ‘Further mandatory information on Course coverage’ section of this document).

Conducting the assessment

Conducted under some supervision and control.

The project will be carried out under open book conditions, but supervised to ensure that the work presented is the candidate’s own work.

The teacher/lecturer may give learners limited guidance to help them progress through each stage of the project; however, the learners are expected to work independently.

The project is designed to discriminate between candidates, and therefore would be expected to provide a wide range of marks. Stronger candidates should be able to complete the project successfully with minimal support and guidance. Weaker candidates may not be able to complete all aspects of the project to a satisfactory standard.

Further mandatory information on Course coverage

The following gives details of mandatory skills, knowledge and understanding for the Advanced Higher Computing Science Course. Course assessment will involve sampling the skills, knowledge and understanding. This list of skills, knowledge and understanding also provides the basis for the assessment of Units of the Course.

The Course assessment will require learners to draw on and apply knowledge of any of the concepts listed below. This table should be read in conjunction with the descriptions of the project and question paper.

The following mandatory generic concepts and vocabulary may be applied to both software design and development and information system design and development.	
Project planning and management	Description and exemplification of iterative project lifecycle: ♦ research: feasibility study, user surveys ♦ planning: scheduling, resources, Gantt chart ♦ analysis of: — user and business requirements — scope, constraints — functional and operational requirements ♦ requirement specifications (for end-users and technical team) ♦ design: — system modelling (data, process) — human computer interaction (user-centred) ♦ implementation: build, integration, deployment ♦ testing: — component, integrative, beta (acceptance), final — usability — accessibility — de-bugging techniques ♦ evaluation: usability, efficiency, effectiveness (goal-tracking), reliability, robustness ♦ maintenance: perfective, corrective, adaptive
Design notations and development methodologies	UML (including class diagrams, use case) pseudocode wire framing data dictionary other appropriate design notations problem decomposition iterative prototyping other contemporary methodologies
Programming paradigms	object-oriented (object, method, property, class, sub-class, encapsulation, inheritance, instantiation) imperative (variables, sequence, selection, iteration, modularity) concurrent (multiple threads, coordination)

Data types and structures	♦ simple data types ♦ structured data types, including: — arrays of records — arrays of objects — 2-D arrays — linked lists — queues — stacks
Standard algorithms	♦ binary search ♦ sort algorithms (selection using two lists, insertion, bubble) ♦ analysis of other algorithms of similar complexity ♦ benefits of other sort algorithms (including quicksort)
Computational constructs	♦ reading and writing data to and from existing files and databases, including inserting, amending and deleting data ♦ form structure using HTML, form processing using server-side script ♦ recursion ♦ SQL operations, including: — DDL > create statement — DML > insert, update and delete statements > select statement (from, where, order by and group by clauses), aggregate functions (count, sum, min, max, avg) and equi-joins between tables
Implications: legal, ethical, environmental economic and social	♦ intellectual property rights (including patent) ♦ storage of user data ♦ increasing use and power of intelligent systems ♦ energy (data centres, low carbon equipment) ♦ online marketing (web, e-mail, text) ♦ analytics ♦ cyber security risks and precautions ♦ tracking, privacy, online safety ♦ social media ♦ implications of ‘big data’

Administrative information

Published: April 2016 (version 2.1)

History of changes to Course Assessment Specification

Version	Description of change	Authorised by	Date
2.0	Additional information provided within the 'Added value' section. Clarification of the requirements for the Course project and additional information regarding the use of SQA standardised reference language within the question paper. Duration of examination increased to 2 hours. Changes made to the concept tables in 'Further mandatory information on Course content' section.	Qualifications Development Manager	April 2015
2.1	Reference to the Question Paper Brief added to the 'Structure and coverage of the Course assessment' section.	Qualifications Manager	April 2016

Note: You are advised to check SQA's website (**www.sqa.org.uk**) to ensure you are using the most up-to-date version of the Course Specification.