Higher
Course Assessment
Specification

Higher Computing Science Course Assessment Specification (C716 76)

This edition: August 2016, version 1.5

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: Higher Computing Science

SCQF level: 6 (24 SCQF credit points)

Course code: C716 76

Course assessment code: X716 76

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 — Question Paper	90 marks
Component 2 — Assignment	60 marks
Total marks	**150 marks**

This Course includes six 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 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 Units, learners will develop software and information system design and development skills, and knowledge and understanding of advanced computing science concepts and processes in a variety of contexts.

This added value consists of the following:

To achieve success in the Course, learners must show that they can **apply** knowledge and skills developed through the Course to solve problems, in both practical and theoretical contexts.

The assignment requires learners to demonstrate aspects of challenge and application in a practical context. Learners will **apply** knowledge and skills from the Course 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 question paper, and an assignment. The question paper will have two Sections.

Component 1 — 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:

- applying computational thinking to understand problems across a range of contexts
- analysing problems with some complex aspects within computing science across a range of contemporary contexts
- the ability to communicate how a program works in technical detail
- communicating understanding of advanced concepts related to software design and development, and information system design and development, clearly and concisely using appropriate terminology
- understanding of the legal, environmental, economic and social impact of contemporary computing technologies
- applying computing science concepts and techniques to create solutions across a range of contexts

The question paper will have 90 marks out of a total of 150 marks. This is 60% of the overall marks for the Course assessment.

Approximately 50% of the marks will be awarded for questions related to software design and development. These will include questions sampling from the following areas:

- contemporary languages and environments
- computational constructs and concepts
- explaining code
- writing code – with algorithm specification
- data types and structures
- software development – design, development methodologies, testing, documentation
- low-level operations and computer architecture
- contemporary developments and trends

Approximately 50% of the marks will be awarded for questions related to information system design and development.

These will include questions sampling from the following areas:

- database design, structures, user interface, links and operations
- website design, structures, user interface, links and coding
- media types, including file size calculations and compression techniques
- information system development – purpose, functionality, user interface, testing
- technical implementation (hardware, software, storage, networking/connectivity)

- ♦ security risks, precautions, legal and environmental implications, economic and social impact

However, many concepts are relevant to both of software and information system design and development, so some questions will relate to both of these broad areas.

Questions assessing understanding and application of programming skills will be expressed using SQA reference language, which may include the following terms:

Variable types:	INTEGER, REAL, BOOLEAN, CHARACTER
Structured types:	ARRAY, STRING, RECORD
System entities:	DISPLAY, KEYBOARD
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
Sub-programs:	id(parameters)

Where required, sub-programs may be presented in the following formats:

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

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

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.

Note: The standardised reference language has been developed by the Universities of Glasgow, Strathclyde and Heriot-Watt and a detailed reference guide can be downloaded from the SQA website.

The question paper has two Sections:

Section 1 will have 20 marks and will consist of short answer questions.

This section will give learners an opportunity to demonstrate breadth of knowledge from across the topics specified in the tables provided in the 'Further mandatory information on Course coverage' section at the end of this *Course Assessment Specification.*

Questions in this section will:

- assess breadth of knowledge
- sample across both Units to ensure balance
- require some calculations

Section 2 will have 70 marks and will consist of structured questions.

This section 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 in this section will:

- assess application of understanding with very few questions requiring direct recall of knowledge
- have balanced sampling across the Course
- consist of questions set in meaningful contexts, which require learners to provide some descriptions and explanations, and may involve some calculations
- include some structured questions that draw on understanding from two or more topics; some parts of these questions may require integration (combining understanding from two or more topics)

Questions related to programming will be presented using SQA standardised reference language.

A proportion of marks will be available for more challenging questions, which could require integration, detailed descriptions or explanations, problem-solving, writing code, and/or analysis, comparison and evaluations.

Component 2 — assignment
The purpose of the assignment is to assess practical application of knowledge and skills from the Units to develop a solution to an appropriately challenging computing science problem. It will assess learners' skills in analysing a problem, designing, implementing and testing a solution to the problem, and reporting on that solution.

The assignment will have 60 marks (40% of the total mark).

Time will be required for:

- preparation for the assignment, which could include considering exemplar assignments and practising required skills
- carrying out the stages of the assignment, with assessor guidance and support
- assessing the process and completed solution

The assignment should clearly demonstrate application of knowledge and skills, at an appropriate level, related to the design and development of software and information systems (as defined in the 'Further mandatory information on Course coverage' section of this document).

Guidelines for the assignment will include a list of questions/tasks/prompts which will lead learners through the assignment in clear stages.

Marks will be awarded for:

- Analysing the problem
- Building a solution (designing, developing, testing)
- Reporting on the solution

Evidence should include:

- the completed digital solution(s) observed by assessor. Hard copy print-outs (including program listings, screenshots, web page source files, data files or similar as appropriate)
- a record of progress through the assignment (such as an informal electronic log, blog or diary maintained by the learner) including all items of evidence specified within the assessment task
- a report on the solution (in written, electronic and/or oral form)
- ephemeral evidence of learner's work including degree of independence that cannot be judged on the basis of other evidence (detailed assessor observation notes and the completed learner assessment record or equivalent)

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 question paper in 2 hours.

Controlled assessment — assignment

The assignment is:

- set by SQA
- conducted under some supervision and control

and

- 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 SQA.

A bank of assignments will be provided, and there will be choice from this bank.

Conducting the assessment

Conducted under some supervision and control.

The assignment will be carried out under open book conditions, but supervised to ensure that the work presented is the learner's own work.

The assessor may give learners support and guidance to help them progress through each stage of the assignment; where any significant amount of support is provided, this should be reflected in the marks awarded.

The assignment is designed to discriminate between learners, and therefore would be expected to provide a wide range of marks. Stronger learners should be able to complete the assignment successfully with minimal support and guidance. Weaker learners may not be able to complete all aspects of the assignment within a reasonable time, or may require significant assistance, and so would achieve a lower total mark.

Once the assignment has been completed and assessed, it must **not** be returned to the learner for further work to improve their mark.

Further mandatory information on Course coverage

The following gives details of mandatory skills, knowledge and understanding for the 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 the Units of the Course.

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

Component 1 — 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. The question paper Component of Course assessment will require learners to draw on and apply knowledge and understanding of a sample of all the topics listed in **all** of the tables below.	
Software Design and Development	
Languages and environments	♦ Description of the following language types: — low-level — high-level — procedural — declarative — object-oriented
Computational constructs	♦ Description, exemplification and implementation of the following constructs: — parameter passing (value and reference, formal and actual) — the scope of local and global variables — sub-programs/routines, defined by their name and arguments (inputs and outputs), including: ○ functions ○ procedures
Data types and structures	♦ Description, exemplification and implementation of the following data types and structures: — string — numeric (integer and real) variables — Boolean variables — 1-D arrays — records — arrays of records — sequential files (open, create, read, write, close)

Testing and documenting solutions	♦ Description and implementation of constructing a comprehensive test plan for a specific problem. ♦ Description and identification of syntax, execution and logic errors. ♦ Description and exemplification of testing techniques including: — dry runs — trace tables/tools — breakpoints — watchpoints
Algorithm specification	♦ Analysis, description, exemplification and implementation of standard algorithms including: — linear search — find minimum and maximum — count occurrences ♦ Analysis of other algorithms of similar complexity.
Low-level operations and computer architecture	♦ Description of the uses of virtual machines and emulators. ♦ Description and exemplification of the use of binary to represent negative integers using two's complement, including the range of numbers that can be represented using a fixed number of bits. ♦ Description of the relationship between the range and precision of real numbers using floating point representation. ♦ Description of Unicode used to represent characters and its advantage over ASCII. ♦ Description of the advantages and dis-advantages of bit-mapped graphics compared to vector graphics. ♦ Understand that sound is represented in binary and described in terms of sample size (bit depth) and sample rate. ♦ Understand that video is represented as a sequence of still frames and described in terms, for each frame, of: — frame rate — resolution — bit depth ♦ Calculation of storage requirements for uncompressed audio and video. ♦ Describe the trends and implications of Computer architecture including: — multi-core processors — parallel processing ♦ Describe the fetch-execute cycle using the components of computer architecture including: — processor (registers, ALU, control unit) — memory — buses (data, address and control)

Software and Information System Design and Development	
The following mandatory generic concepts and vocabulary may be applied to both software design and development and information system design and development.	
Design notations	♦ Description, exemplification and implementation of pseudocode to solve problems. ♦ Description, exemplification and implementation of entity relationship diagrams. ♦ Exemplification and implementation of data dictionary including name, type, size, required and validation. ♦ Exemplification and implementation of wire-framing.
Development methodologies	♦ Description of the general Iterative phases of the development process: analysis, design, implementation, testing, documentation, evaluation, maintenance. ♦ Description, identification and benefits of Development methodologies including: — rapid application development — top-down/step-wise refinement — Agile methodologies
Contemporary developments	♦ Exemplification of trends in the development of: — software development languages — software development environments — intelligent systems — online systems
User interface	♦ Description of problems with accessibility of computer systems and how they can be overcome including: — vision impairments — hearing impairment — motor and dexterity impairments

Information System Design and Development	
The following mandatory generic concepts and vocabulary may be applicable to a range of information systems types and concepts	
Structures and links (database)	♦ Implementation of relational databases with a minimum of three linked data tables. ♦ Description, implementation and exemplification of compound keys and surrogate keys. ♦ Description, exemplification and identification of entity relationships (one-to-one, one-to-many, many-to-many). ♦ Description and implementation of complex database operations including: — input (forms) — searching/sorting/calculations (queries) — output (reports)
Structures and links (web-based)	♦ Description, exemplification and implementation of the site structure of multi-level web-based information system. ♦ Description and implementation of the page structure of web-based information system including head, title, body. ♦ Description, exemplification and implementation of cascading style sheets. ♦ Understand the composition of meta tags and how they are used in search engine optimisation. ♦ Description and advantages/dis-advantages of dynamic web pages and database-driven website. ♦ Description and exemplification of interactive web pages.
Media types	♦ Description of the difference between lossy and lossless compression. ♦ Description and identification of a number of compression techniques including: — Perceptual coding — audio lossy compression technique — Free Lossless Audio Codec — lossless compression technique — RLE — graphic lossless compression technique — LZW encoding — graphic lossless compression technique — DCT encoding — graphic lossy compression technique — Interframe and Intraframe video compression techniques
Coding	♦ Description, exemplification and implementation of coding to create and modify information systems including the use of: — client-side scripting using javascript mouse events — scripting (database/web pages) ♦ Description of the role of server-side scripting in the generation of dynamic web pages and database-driven websites including: — receiving user input/selection from client device

	— validation of form data — connecting to database server — page generation
Testing	♦ Understand the process and benefits of Beta testing. ♦ Describe the process and benefits of usability testing. ♦ Understand that compatibility issues may occur within information systems including: — sufficient memory and storage requirements — compatibility with the operating system
Purpose, functionality, users	♦ Descriptions of purpose, functions, features and appropriate users of a specific information system. ♦ Description of the interaction of information systems with search engines.
Technical Implementation (hardware requirements)	♦ Description and exemplification of the appropriate hardware required for a specified information system including: — input and output devices — processor type, number and speed (Hz) — memory (RAM, cache)
Technical Implementation (software requirements)	♦ Description of the main functions of an operating system including: — interpreting user commands — file management — memory management — input/output management — resource allocation ♦ Description and comparisons of proprietary v open source software licenses. ♦ Understand the benefits of portability for computer programs and information systems. ♦ Description and exemplification of current trends in operating system design. ♦ Description and exemplification of the appropriate type of software required for a specific information system including: — type of application — operating system
Technical implementation (storage)	♦ Description and benefits of distributed storage. ♦ Description and benefits of offline storage. ♦ Description of the advantages/dis-advantages of Cloud systems compared to local server provision including: — cost — accessibility — maintenance ♦ Description and comparison between public, private and hybrid cloud systems.

	♦ Description of backup systems and strategy including: — schedule — frequency, differential, incremental — media — DAT, DLT, optical — location — on-site, off-site repository, cloud — mirroring (RAID) ♦ Description and exemplification of the appropriate type of storage required for a specific information system including: — type of device — capacity — interface type and data transfer speed ♦ Description and exemplification of current trends in storage systems.
Technical implementation (networking/ connectivity)	♦ Description and exemplification of cloud-based services including: — data storage — mail services — software updates ♦ Description and exemplification of web hosting. ♦ Description and exemplification of current trends in networking and connectivity including: — bandwidth — transmission media — hardware such as hubs, switches and routers ♦ Description and exemplification of the appropriate type of network connection required for a specific information system including: — hardware — transmission media — bandwidth
Security risks	♦ Description, identification and exemplification of spyware including: — Trojans — Adware — tracking cookies ♦ Description and exemplification of DOS (Denial of Service) attacks including: — symptoms — slow performance, inability to access — effects — disruption to users — costs — lost revenue, labour in rectifying fault — type of fault — bandwidth consumption, resource starvation, routing, Domain Name Service(DNS) — reasons — financial, political, personal
Security precautions	♦ Description and exemplification of encryption used to secure transmission of data including use of public and private keys ♦ Description and exemplification of digital certificates and signatures

Legal implications	♦ Description, identification and implications for individuals, businesses and ISP's of the Regulation of Investigatory Powers Act including: — intercepting and monitoring of electronic communications by government bodies — monitoring of employees communications — equipment and services used for surveillance
Environmental implications	♦ Description and implications of the lifetime carbon footprint including: — manufacture of computer systems and peripherals — electricity use during a computer systems lifetime — disposal including re-cycling and extraction of dangerous elements ♦ Description and implications of environmental benefits of computer systems including: — reduction in paper use in offices etc — reduction in manufacturing/transportation due to increased downloading of music and books — reduction in travelling through working from home — intelligent control of heating systems
Economic and social impact	Economic impact: ♦ Description and exemplification of the competitive advantage computer Systems give businesses. ♦ Implications of the global marketplace for Business and customers. ♦ Description and exemplification of business costs involved in the maintainability and scalability of Information systems including: — training — hardware — software — storage — connectivity Social impact: ♦ Comparison between censorship and freedom of speech in relation to the internet. ♦ Exemplification of the safeguards required to ensure privacy when using information systems such as social media sites. ♦ Understand the advantages of global citizenship. ♦ Exemplification of advantages and dis-advantages of online communities.

Component 2 — Assignment
The purpose of the assignment is to assess practical application of knowledge and skills from across the Course to develop a solution to an appropriately challenging computing science problem. It will assess learners' skills in analysing a problem, designing, implementing (developing) and testing a solution to the problem, and reporting on that solution. The assignment Component of the Course assessment will require learners to apply knowledge and understanding of a sample of the topics listed in **all** of the tables above.

Administrative information

Published: August 2016 (version 1.5)

History of changes to Course Assessment Specification

Version	Description of change	Authorised by	Date
1.1	Incorrect reference to Units instead of Course changed on page 3. Additional information has been added to the Structure and coverage of the Course assessment section regarding the question paper; clarification of allocation of marks for the assignment and evidence required, and addition of pseudocode terms. Component headings with supporting information have been included in the Further mandatory information on Course coverage section. This has been split into three tables, merging topics covered by both subjects, and content clarified in various topic areas. House style corrections made throughout.	Qualifications Development Manager	April 2014
1.2	Minor clarifications to language specification (functions) on pages 6 and 7, and records on page 10; “standardised pseudocode” now referred to as ‘reference language’.	Qualifications Manager	August 2014
1.3	Minor amendment to the ‘Setting, conducting and marking the assessment’ section — ‘should’ changed to ‘must’. Data Protection Act added to the requirements for Course coverage for Information System Design and Development Unit.	Qualifications Manager	June 2015
1.4	Reference to the Question Paper Brief added to the ‘Structure and coverage of the Course assessment’ section. Depth and clarification added to the mandatory Course content.	Qualifications Manager	April 2016
1.5	The following changes were made to the 'Further mandatory information on Course coverage' table: ‘DTL’ changed to ‘DLT’ when referring to 'Technical implementation (storage)'. ‘Sample size’ changed to ‘Sample size (bit depth)’ in the ‘Low-level operations and computer architecture’ section. Correction to remove the second ‘constructing a’ from the Testing and documenting solutions section. Small change to bulleted layout in the ‘Technical Implementation (software requirements)’ section.	Qualifications Manager	August 2016