National Unit Specification: general information

UNIT **Developing a Software Solution (Advanced Higher)**

NUMBER DM43 13

COURSE Computing (Advanced Higher)

SUMMARY

This Unit is designed to develop the ability to analyse a complex computing problem and to design, implement and test a software based solution.

It is suitable for candidates who have a background in software development up to Advanced Higher level, and is designed as a preparation for undertaking the Coursework Project for the Advanced Higher Computing Course.

OUTCOMES

1. Demonstrate knowledge and understanding of the analytical approach to the development of a software solution to a computing problem.
2. Demonstrate practical skills by analysing, designing, implementing and testing a software solution to a computing problem.

RECOMMENDED ENTRY

While entry is at the discretion of the centre, candidates would normally be expected to have attained one of the following, or equivalent:

- Higher Computing
- *Software Development* (Advanced Higher) Unit

Administrative Information

Superclass: CB

Publication date: April 2005

Source: Scottish Qualifications Authority

Version: 1

National Unit Specification: general information (cont)

UNIT Developing a Software Solution (Advanced Higher)

CREDIT VALUE
1 credit at Advanced Higher (8 SCQF credit points at SCQF level 7).*

**SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from Access 1 to Doctorates.*

CORE SKILLS

Core Skills Components for this Unit	Critical Thinking (Higher) Planning and Organising (Higher).

National Unit Specification: statement of standards

UNIT Developing a Software Solution (Advanced Higher)

Acceptable performance in this Unit will be the satisfactory achievement of the standards set out in this part of the Unit specification. All sections of the statement of standards are mandatory and cannot be altered without reference to the Scottish Qualifications Authority.

OUTCOME 1

Demonstrate knowledge and understanding of the analytical approach to the development of a software solution to a computing problem.

Performance criteria

(a) The terminology of the analytical approach is used appropriately.
(b) Descriptions and explanations are technically accurate and concise.
(c) The stages of the software development process are exemplified clearly.

Evidence requirements

Written or oral evidence that the candidate can describe and explain the analytical approach to the development of a software solution to a computing problem. Evidence should be obtained using questions in a closed-book test, under supervision lasting no more than 20 minutes. The test must sample across the range of the content (see Computing (Advanced Higher) Course content) relevant to this Unit.

(The content statements are also reproduced for convenience as a table in the support notes for this Unit)

The standard to be applied is illustrated in the National Assessment Bank items available for this Unit.

If a centre wishes to design its own assessments for this Unit, they should be of a comparable standard.

National Unit Specification: statement of standards (cont)

UNIT Developing a Software Solution (Advanced Higher)

OUTCOME 2
Demonstrate practical skills by analysing, designing, implementing and testing a software solution to a computing problem.

Performance criteria

(a) Analysis of the problem is clear and concise.
(b) Appropriate information resources are used effectively.
(c) Design of solution is clearly expressed and complete.
(d) The solution demonstrates use of complex programming techniques.
(e) Hardware is used effectively and efficiently to implement a solution.
(f) Software is used effectively and efficiently to implement a solution.
(g) Test plan is clear and appropriate, and testing is carried out systematically.

Evidence requirements

Written or oral evidence that the candidate can analyse a complex computing problem, and design, implement and test a software solution to this problem. The evidence should be generated by the candidate during the development of a software solution to a real computing problem. This evidence will take the form of a record of work.

The record of work need not comprise a formal report but must provide evidence that the candidate has completed each of the following stages of the development process to the level defined by the Performance Criteria and the content statements (see Computing (Advanced Higher) Course content):

- analysis
- design
- implementation
- testing.

The candidate will be expected to use appropriate books, notes and on-line help whilst completing the record of work.

(The content statements are also reproduced for convenience as a table in the support notes for this Unit)

The standard to be applied is illustrated in the National Assessment Bank items available for this Unit.

If a centre wishes to design its own assessments for this Unit, they should be of a comparable standard.

National Unit Specification: support notes

UNIT Developing a Software Solution (Advanced Higher)

This part of the Unit specification is offered as guidance. The support notes are not mandatory.

While the exact time allocated to this unit is at the discretion of the centre, the notional design length is 40 hours.

GUIDANCE ON THE CONTENT AND CONTEXT FOR THIS UNIT

Candidates are expected to be able to conduct independent research at this level and it is anticipated that there will be limited teacher/lecturer exposition of the underlying principles and concepts. Candidates will use a wide range of sources of information including paper based and electronic.

It is expected that the candidate will apply and extend their knowledge of software development by solving a problem which offers an appropriate depth of complexity.

The content for this Unit is detailed below (and also in the National Course Specifications: Course details.)

Content Statements: Developing a software solution
Description and exemplification of a problem specification. Description and exemplification of the elements of the analysis stage: ♦ statement of the requirements ♦ identification of scope and boundaries of the problem ♦ identification of functional requirements
Description and exemplification of the elements of a project plan: ♦ identification of sub-tasks ♦ setting a realistic time-scale ♦ application of appropriate project management technique
Description of the need to: ♦ consider and compare possible strategies using clearly specified criteria ♦ select and justify a strategy
Description and exemplification of aspects of a good user interface.
Description and exemplification of the elements of the testing stage of the process: ♦ creation of a test plan ♦ creation of test data ♦ systematic testing ♦ user questionnaire ♦ summary of results ♦ rectifying errors and bugs

National Unit Specification: support notes (cont)

UNIT Developing a Software Solution (Advanced Higher)

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

Candidates will require individual access to appropriate computer hardware and software throughout this Unit.

The two Outcomes should be delivered in an integrated way rather than sequentially. The stages of the software development process should be taught and then applied to an appropriate practical task.

The amount of time spent on each area of content will vary, depending on the teaching methodology used and the ability and prior experience of the candidates. However, the following times are suggested as a rough guide:

problem specification	2 hours
project planning	2 hours
analysis	4 hours
design	4 hours
implementation	20 hours
testing	4 hours

1½ hours should be set aside to:
- administer the Outcome 1 test
- gather evidence for Outcome 2

A further 2½ hours is allowed for remediation and re-assessment if required.

If the Unit is delivered as part of the Advanced Higher Computing Course, candidates will extend the work of this Unit into the Coursework element of external assessment, by producing a formal report on the problem and its solution.

The following guidance is offered to help centres to determine whether potential computing problems are at an appropriate level for this Unit.

Appropriate problems for this Unit should:
- allow the candidate to demonstrate knowledge and understanding gained during completion of the *Software Development Systems* Unit at Advanced Higher level
- provide a suitable level of difficulty and complexity, appropriate to Advanced Higher level and allowing the candidate to develop a software solution to a computing problem that requires the development of complex algorithms
- allow sufficient scope for development, enabling candidates to demonstrate all knowledge and skills required for assessment purposes
- be possible using hardware and software resources available in the teaching centre
- be possible within the 40 hours available for completion of this Unit.

Suitable problems might include development of:
- a simulation of a card game with a graphical user interface and an element of artificial intelligence developed using an appropriate high level programming language
- an encryption/decryption program involving file handling and complex key algorithms
- an implementation of a natural language interpreter using a declarative language and based on a recognised grammar and parsing technique

- a computer aided learning package which demonstrates an assembly language emulator for teaching the *Computer Systems* Unit and incorporating a number of commonly used op-codes and addressing modes
- a network application to allow peer-to-peer chat facilities
- a website which incorporates advanced techniques such as the development of user interactivity, form filling or a front end to a database system making extensive use of PHP, Java or PERL
- an expert system developed without the aid of an expert system shell.

It is important that the problem chosen allows scope for the candidate to demonstrate appropriate complex programming techniques. The following examples, while similar to those listed above, are unlikely to provide sufficient depth and challenge to be suitable for this Unit:

- a simulation of a card game using only code modules easily available in the public domain
- an encryption/decryption program using only a simple scrambling algorithm such as one which would be within the reach of a Higher candidate
- an implementation of a natural language interpreter based on a very limited vocabulary and simplistic approach to grammar rules
- a computer aided learning package using only simple multimedia authoring tools
- a website which does not include advanced techniques such as scripting
- an expert system for a careers database which does not include advanced rules or other constructs available within an expert system shell.

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

National Assessment Bank tests have been created specifically to assess Outcome 1 of the Unit. This assessment consists of a closed book test, and must be conducted under supervision. In order to gain success in this Outcome, the candidate must achieve at least the cut-off score for the test. If a centre wishes to design its own assessments for this Unit, they must be of a comparable standard.

Outcome 2 requires the candidate to demonstrate practical skills while using contemporary hardware and software to implement and test the software solution to a computing problem. The skills will normally be demonstrated by the candidate during the development of the software solution, rather than as separate formal assessment activities. The candidate will be allowed access to books, notes and on-line help while completing the tasks. The practical skills should be demonstrated in the context and at a level defined by the content statements (see Advanced Higher Course content).

The assessment of Outcome 2 is based on a record of work produced by the candidate. The record of work will normally be generated by the candidate during the development of a software solution, rather than as separate formal assessment activities. The evidence will be generated by the candidate as a series of notes, annotated diagrams, screen shots and notes on testing. A formal report is not required for Unit assessment.

The candidate will be allowed access to books, notes and on-line help while completing the record of work.

To gain success in Outcome 2, the candidate must demonstrate practical skills in each of the following stages of the software development process:

- analysis
- design
- implementation
- testing

National Unit Specification: support notes (cont)

UNIT Developing a Software Solution (Advanced Higher)

All evidence must be retained by the centre. The assessment of this Unit is subject to central moderation by SQA.

A pro-forma observation checklist for Outcome 2 is provided in the National Assessment Bank materials.

CANDIDATES WITH ADDITIONAL SUPPORT NEEDS

This Unit Specification is intended to ensure that there are no artificial barriers to learning or assessment. The additional support needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments or considering alternative Outcomes for Units. For information on these, please refer to the document *Guidance on Assessment Arrangements for Candidates with Disabilities and/or Additional Support Needs* (SQA, 2004).