

Higher National Unit Specification

General information for centres

Unit title: Software Development: Structured Programming

Unit code: DH3E 35

Unit purpose: This Unit is designed to develop a broad knowledge of the concepts, principles, boundaries and scope of software development using a structured programming language. These will be reinforced by developing the practical skills required in using the structures and features of a procedural programming language in the creation of software solutions to problems. It forms part of an HN Computing group award programme, although it can also be used as a stand-alone Unit by candidates wishing to acquire and develop programming skills using a procedural language.

On completion of this Unit, the candidate should be able to:

1. Use programming techniques to develop program modules.
2. Implement a solution from a design.
3. Test the completed product.
4. Create technical and user documentation.

Credit value: 2 HN Credits at SCQF level 8: (16 SCQF credit points at SCQF level 8*)

**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.*

Recommended prior knowledge and skills: Access to this Unit will be at the discretion of the Centre, however it is recommended that candidates should have achieved the core skill of Problem Solving at Intermediate 1 level before taking this Unit. It is also recommended that candidates have prior experience of using computer systems or studied a programming language at NQ level. This may be evidenced by possession of HN Unit DH33 34 *Computer Operating Systems 1*, or HN Unit D75X 34 *Information Technology: Applications Software 1*. However, it is expected that a candidate is either studying this unit in conjunction with or following the completion of the HN Unit DH35 34 *Computing: Planning*. Alternatively, candidates should have considerable practical work experience and some appreciation of the role of program design and implementation.

Core skills: There may be opportunities to gather evidence towards core skills in this Unit, although there is no automatic certification of core skills or core skills components.

General information for centres

Context for delivery: If this Unit is delivered as part of a group award, it is recommended that it should be taught and assessed within the subject area of the group award to which it contributes.

Assessment: The Unit must be assessed by means of a single project covering all four Outcomes with the exception of part of Outcome 2 that is closed book. The achievement requirements are inherent in the evidence requirements. Assessors must assure themselves of the authenticity of each candidate's submission.

All outcomes are open book with the exception of part of Outcome 2.

Some of the evidence requirements may be produced using e-assessment. This may take the form of e-testing (for knowledge and understanding) and/or e-portfolios (for practical abilities). There is no requirement for you to seek prior approval if you wish to use e-assessment for either of these purposes so long as the normal standards for validity and reliability are observed. Please see the following SQA publications for further information on e-assessment: (1) "SQA Guidelines on Online Assessment for Further Education" (March 2003) and (2) "Assessment & Quality Assurance in Open & Distance Learning" (Feb. 2001).

If a centre is presenting the closed book part of Outcome 2 on-line the following assessment methods, where appropriate, may be selected –

- ◆ Multiple-choice
- ◆ Drag and drop
- ◆ Multiple response
- ◆ Mix and match
- ◆ A combination of the above

It is expected that the questions will be of the multi-choice variety. Centres may consider the use of alternative questions types, particularly if using Computer Assisted Assessment approaches. However, care should be taken that the questions are valid and at an appropriate level. The use of simple true/false question responses is unlikely to achieve this.

Note – Credit transfer should not be given between the outcomes in the following HN Units:

DH3C 35: Software Development: Object Oriented Programming
DH34 35: Software Development: Event Driven Programming
DH3E 35: Software Development: Structured Programming
DH30 35: Software Development: Applications Development
DH32 35: Software Development: Developing for the World Wide Web

Higher National Unit specification: statement of standards

Unit title: Software Development: Structured Programming

Unit code: DH3E 35

The sections of the Unit stating the Outcomes, knowledge and/or skills, and evidence requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the knowledge and/or skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

Outcome 1

Use programming techniques to develop program modules

Knowledge and Skills

- ◆ Declaring and initialising variables
- ◆ Using operators and predefined functions
- ◆ Using standard input/output commands
- ◆ Implementing control structures
- ◆ Using parameter passing mechanism to transfer values between modules
- ◆ Defining data structures
- ◆ Accessing and manipulating data structures and input/output file operations

Evidence Requirements

The design to be implemented is either derived in the HN Unit: DH35 34 *Computing: Planning* or provided by the assessor. The candidate will implement selected data structures and control structures depending upon the design.

Achievement in this Outcome will be evidenced by the production of code modules. The coding must encompass all of the above features, and must meet the given design specifications using at least one example of each of the features listed below. The evidence must demonstrate the appropriate use and implementation of these features to meet the requirements of both the given project brief and the problem specifications.

Specifically, the following features **must** be demonstrated in the candidate's program listing:

- ◆ Variables are appropriately defined, declared and initialised
- ◆ Arithmetic or logical operators are correctly used
- ◆ Appropriate mathematical, character, or date (or time) functions are correctly used
- ◆ Control structures, sequence, selection and iteration are correctly used
- ◆ Appropriate parameter passing is correctly used
- ◆ Modules are internally documented to organisational standards
- ◆ Appropriate error and contingency handling requirements are correctly used.

Higher National Unit specification: statement of standards (cont)

Unit title: Software Development: Structured Programming

The data structures used should include standard and user-defined types.

This assessment is open book.

Assessors must assure themselves of the authenticity of each candidate's submission.

Outcome 2

Implement a solution from a design

Knowledge and Skills

- ◆ Identify language components from the design
- ◆ Use tools and techniques to create software modules
- ◆ Implement error handling
- ◆ Demonstrate an understanding of programming techniques and concepts

Evidence requirements

Each candidate will undertake **two separate** assessment tasks to cover the evidence required.

Outcome - Assessment Task 1

Each candidate will need evidence to show that s/he can, with reference to the design supplied or produced, implement the completed solution. The tools and techniques available within the programming language should be utilised. A candidate's response can be judged to be satisfactory where the evidence provided is sufficient to meet the requirements for each item by showing that the candidate is able to:

- ◆ Supply programming code
- ◆ Implement all required software components and data structures

In the assessment the candidate will be required to produce documentation based on the derived specification. The documentation may be supplemented by additional evidence to ensure coverage of all aspects of the evidence requirements. The format of the documentation must follow organisational standards as defined by the Centre and is a part of the mandatory evidence requirements for this Unit.

This assessment is open book.

Assessors must assure themselves of the authenticity of each candidate's submission.

Higher National Unit specification: statement of standards (cont)

Unit title: Software Development: Structured Programming

Outcome - Assessment Task 2

Evidence for all the knowledge and/or skills in this Outcome will be assessed using a representative sample of 20 questions covering programming techniques and concepts. These questions can be presented from the following:

- ◆ Multiple choice
- ◆ Drag and drop (on-line delivery only)
- ◆ Multiple response
- ◆ Mix and match
- ◆ A combination of the above

The questions presented must change on **each** assessment occasion.

Assessors must ensure that **all questions** should be at SCQF Level 8.

Assessment must be undertaken in supervised conditions and is closed book. A candidate should complete this assessment within one hour. Candidates may not bring to the assessment event any notes, textbooks, handouts or other material.

Candidates must answer at 60% of the questions correctly.

Assessment guidelines

Outcome - Assessment Task 1

The achievement requirements are inherent in the evidence requirements.

Outcome - Assessment Task 2

There is an opportunity for a candidate to be assessed on-line subject to meeting the prescribed assessment conditions.

If a centre is presenting this assessment on-line the following assessment methods, where appropriate, may be selected –

- ◆ Multiple-choice
- ◆ Drag and drop
- ◆ Multiple response
- ◆ Mix and match
- ◆ A combination of the above

Higher National Unit specification: statement of standards (cont)

Unit title: Software Development: Structured Programming

Outcome 3

Test the completed product

Knowledge and Skills

- ◆ Implementing a test plan using a defined strategy
- ◆ Maintaining a test documentation
- ◆ Evaluating results of test runs
- ◆ Amending coding as necessary

Evidence Requirements

The test plan to be implemented is either derived in the HN Unit: DH35 34 *Computing: Planning* or provided by the assessor. This Outcome will be assessed by the production of completed test documentation recording both the expected results of the test data and the actual results. The bank of test data should be sufficient to thoroughly test the implemented solution in scope and range.

The candidate will be expected to record and evaluate the results of the test runs. Where there are discrepancies between the expected results and the actual results, the coding must be amended and corrected accordingly.

Since the implemented solution should meet the requirements of the project brief, the candidate must produce a complete and error-free program solution to meet the project brief and problem specifications.

This assessment is open book.

Assessors must assure themselves of the authenticity of each candidate's submission.

Outcome 4

Create technical and user documentation

Knowledge and skills

- ◆ Preparing technical documents
- ◆ Preparing user documents

Higher National Unit specification: statement of standards (cont)

Unit title: Software Development: Structured Programming

Evidence requirements

This Outcome will be evidenced by the candidate producing two documents as follows:

- ◆ Technical Manual containing
 1. The project brief
 2. Specification document
 3. Design document
 4. Program code listing
 5. Test documents

- ◆ User Manual containing
 1. Installation requirements and guide
 2. User guide procedures for start-up and use of the software and which includes a list of error messages and their resolution

This assessment is open book.

Assessors must assure themselves of the authenticity of each candidate's submission.

Administrative Information

Unit code: DH3E 35

Unit title: Software Development: Structured Programming

Superclass category: CB

Original date of publication: June 2004

Version: 02 (April 2007)

History of changes:

Version	Description of change	Date
02		27/4/07

Source: SQA

© Scottish Qualifications Authority 2004, 2007

This publication may be reproduced in whole or in part for educational purposes provided that no profit is derived from reproduction and that, if reproduced in part, the source is acknowledged.

SQA acknowledges the valuable contribution that Scotland's colleges have made to the development of Higher National qualifications.

Additional copies of this Unit specification can be purchased from the Scottish Qualifications Authority. Please contact the Customer Contact Centre for further details, telephone 0845 279 1000.

Higher National Unit specification: support notes

Unit title: Software Development: Structured Programming

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 80 hours.

Guidance on the content and context for this Unit

This Unit is intended as an introduction to the basic principles and procedures involved in developing robust, reliable, efficient and maintainable software using a recognised structured methodology. Its aim is to acquire competence in the development of software solutions using the program language syntax and constructs required to reinforce software development. The concepts of good practice, such as internal documentation, should be stressed throughout. The Unit should provide a foundation of good programming skills on which the more advanced programming Units such as – HN Unit *Software Development: Array Data Structures* followed by any of the following: *Software Development: Linked Data Structures* or *Software Development: Object Oriented Collections* or *Software Development: Advanced Programming*.

It forms part of an HN Computing group award programme and should be delivered within the context of the group award. It can also be delivered as a stand-alone Unit by candidates wishing to acquire, and develop, programming skills using a structured programming language.

As an introductory Unit in programming on an HN group award, the context and examples used should be both relatively simple and within the context of the candidate's vocational experience. This would permit the candidate to concentrate more on the programming techniques involved, rather than trying to understand the initial problem.

By the end of the Unit, the candidate should have achieved a good foundation in the skills required for developing reliable, robust and efficient program designs solve business problems using a structured programming language.

It is likely that the Unit will be delivered in the first year of a full-time HN Computing group award.

The Unit is capable of being delivered on its own, but could also be delivered in conjunction with the HN Unit DH35 34 *Computing: Planning*. The stages of the determined design methodology may be presented intermittently with programming principles from this Unit, with simple practical exercises in design techniques being presented and these designs being implemented in the chosen structured programming language.

For assessment purposes, the implementation should be based on a design. If the Unit is to be delivered in conjunction with the HN Unit DH35 34 *Computing: Planning*, then the design will be the product of the assessments for that Unit. If the Unit is being delivered as a stand-alone Unit, then the candidate will be issued with a design for implementation.

Higher National Unit specification: support notes (cont)

Unit title: Software Development: Structured Programming

This Unit is intended as an introduction to the basic principles and procedures involved in developing robust, reliable, efficient and maintainable software using a recognisable structured methodology. It has been contextualised for the HNC Games Development award, where the emphasis is on encouraging learner creativity and design flair in a digital entertainment domain. All of the standard good practice criteria are also stressed throughout the Unit, but with a games development flavour. In the production of videogames the learners acquire competence in the development of software solutions using the program language syntax and constructs required to reinforce software development.

It forms part of the HNC Games Development Group Award and should be delivered within the context of the Group Award.

Instruments of assessment

All of the instruments of assessment are based on a single project that the student works through to achieve the four Outcomes. The exception is part of Outcome 2 as this is assessed by a closed-book test.

The project should be a graphics-based game. For example, the game could be a card, puzzle, 2D platform, or 3D-based type, but must involve some form of file I/O to satisfy LO1 requirements. Game design specifications should be supplied for the game. Once the game has been implemented, using a recognised game programming development environment (Dark Basic/C++, etc). The candidate must then systematically test the game using recognised testing techniques, and finally produce a technical manual and user manual.

All of the instruments of assessment are open-book except for the test in part of Outcome 2.

Assessors must ensure that all candidate submissions for the project-based assessments are the work of the candidate. The level of work required for the assessments is inherent in the evidence requirements.

The closed-book part of Outcome 2 will be assessed via a twenty question multiple choice test. The reassessment test will be based on entirely different questions to the first assessment test.

Higher National Unit specification: support notes (cont)

Unit title: Software Development: Structured Programming

Guidance on the delivery and assessment of this Unit (cont)

The assessor must ensure that candidate evidence is being produced at SCQF Level 8. The implementation of the design requires the candidate to consider and select **in each** of the following areas:

- ◆ appropriate data structures including user defined data types
- ◆ appropriate data structure manipulation including input, output and linear search
- ◆ sequential file manipulation for input and output.

The principles of good practice, such as internal documentation and code indentation, should be included throughout.

After introducing the steps involved in developing a solution to typical problems, candidates should be presented with a series of practical exercises to illustrate the features of the programming language as they are introduced.

Outcome 2 consists of two separate tasks, one practical and one closed book. The closed book assessment consists of 20 questions that should cover programming techniques and concepts covered in the implementation of the design.

Regardless of delivery mode, a single and coherent case study approach is recommended for assessment, with the model broken down into identifiable and assessable stages allowing the tutor the opportunity for regular monitoring of the candidate's progress and allowing time for intermittent remediation.

Testing should involve the candidate comparing the expected and the actual results, evaluating the differences and amending the code as necessary.

The test data may be the product of HN Unit DH35 34 *Computing: Planning* where the two Units are being delivered in conjunction. If this programming Unit is being delivered on a stand-alone basis, the bank of test data may be supplied by the assessor.

During the learning process, part-completed code modules may be offered to the candidate for completion of the key syntax and language features.

Higher National Unit specification: support notes (cont)

Unit title: Software Development: Structured Programming

Open learning

If this Unit is delivered by open or distance learning methods, additional planning and resources may be required for candidate support, assessment and quality assurance.

A combination of new and traditional authentication tools may have to be devised for assessment and re-assessment purposes. For further information and advice, please see *Assessment and Quality Assurance for Open and Distance Learning* (SQA, February 2001 — publication code A1030).

Candidates with disabilities and/or additional support needs

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. Further advice can be found in the SQA document *Guidance on Assessment Arrangements for Candidates with Disabilities and/or Additional Support Needs* (www.sqa.org.uk).

General information for candidates

Unit title: Software Development: Structured Programming

In this Unit, you will acquire the knowledge and learn the skills necessary to enable you to develop software from a design.

To achieve this, you will use a range of tools that will assist you in developing your ideas towards robust, reliable and efficient software.

This will involve the following areas of learning.

- Using the features of a structured programming language, you will implement a software solution from a design. You will learn how to apply a structured methodology to help you to implement working software, which meets the user's needs. Your understanding and grasp of these skills and techniques will be reinforced throughout with practical exercises.
- Using a test plan, you will test your software to ensure it works correctly and meets the user requirements. You will be required to amend any errors in your code in order to achieve a robust, reliable and efficient working program.
- Create technical and user manuals.

Within Outcome 2 you will be required to answer a number of questions in a closed book assessment situation.