

National Unit Specification: general information

UNIT Computing: Programming in a High-level Language —
Fundamentals (SCQF level 4)

CODE F1K0 10

SUMMARY

This Unit is designed to introduce the concept of computer programming by identifying the various components needed to construct simple programs using a high level language. At the end of the Unit the candidates will gain sufficient knowledge to create a simple software program for a given problem. This Unit is language independent and should provide a framework for other language specific programming Units.

This Unit is aimed at candidates who have no previous experience of computer programming.

OUTCOMES

- 1 Identify the requirements for creating a computer program.
- 2 Identify the various components used in constructing computer programs.
- 3 Use structured techniques in program design.
- 4 Create a software program for a given problem.

RECOMMENDED ENTRY

While entry is at the discretion of the centre, it would be beneficial if candidates possessed basic IT skills. This may be evidenced by possession of:

D01D 09 *Information Technology (Access 3)*

or equivalent qualifications or experience.

Administrative Information

Superclass: CB

Publication date: June 2007

Source: Scottish Qualifications Authority

Version: 01

© Scottish Qualifications Authority 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.

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

National Unit Specification: general information (cont)

UNIT **Computing: Programming in a High-level Language — Fundamentals**
(SCQF level 4)

CREDIT VALUE

1 credit at intermediate 1 (6 SCQF credit points at SCQF level 4*)

**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

While there is no automatic certification of Core Skills in Unit there may be opportunities for developing aspects of Core Skills.

National Unit Specification: statement of standards

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

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

Identify the requirements for creating a computer program.

Performance Criteria

- (a) Identify software and hardware needed to generate computer programs.
- (b) Identify the different stages of the programming process.
- (c) Identify various programming development paradigms commonly used.
- (d) Identify the differences between compiled and interpreted programming languages.

OUTCOME 2

Identify the various components used in constructing computer programs.

Performance Criteria

- (a) Identify the differences between input and output data.
- (b) Identify simple/primitive data types.
- (c) Identify the advantages of modular programming using subroutines/procedures.
- (d) Identify different types of control structures.
- (e) Identify the various types of operators used in programming.

OUTCOME 3

Use structured techniques in program design.

Performance Criteria

- (a) Identify various structured techniques.
- (b) Describe the advantages of structured techniques.
- (c) Implement a program design for a given problem.

OUTCOME 4

Create a software program for a given problem.

Performance Criteria

- (a) Use simple/primitive data types.
- (b) Use control structures and operators.
- (c) Write and run a computer program for a given problem.

National Unit Specification: statement of standards (cont)

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

EVIDENCE REQUIREMENTS FOR THIS UNIT

Evidence is required that candidates have achieved all Outcomes and Performance Criteria.

A candidate is encouraged to use the Internet in any research etc, however, the evidence produced must be the candidate's own work and words. Assessors should ensure themselves of the authenticity of candidate's evidence.

Written and/or oral recorded evidence is required which demonstrates that the candidate has achieved the requirements of all of the Outcomes and Performance Criteria to show that the candidate has appropriate knowledge and understanding of the content of this Unit.

For Outcomes 1 and 2 the evidence produced will take the form of written and/or oral recorded evidence. For Outcomes 3 and 4 the evidence will consist of product and an assessor observation checklist which provides the evidence that the candidate has achieved all of the Performance Criteria.

Evidence for Outcomes 1 and 2 will include:

- ◆ The stages of the programming process, ie analysis, design, coding, implementation, testing, maintenance and documentation
- ◆ **Three** programming development paradigms commonly used in industry, eg structured, object oriented, and event driven
- ◆ The **three** control structures, ie sequence, selection, and repetition
- ◆ **Three** operators, ie arithmetic, assignment and comparison (logical) operators
- ◆ **Two** differences between input and output data
- ◆ **Two** different simple/primitive data types used in programming, eg integers, real and characters
- ◆ **Two** advantages of modular programming using subroutines/procedures

Product evidence for Outcomes 3 and 4 will consist of:

- ◆ a program design using graphical and non-graphical structured techniques
- ◆ code listing
- ◆ snapshots of the generated output

An assessor observation checklist is required to verify the candidate's completion of all of the following practical tasks:

- ◆ Appropriate use of program generation and execution environment, including at least two of the following: integrated environments, text editors, compilers and browser
- ◆ Declaration and assignment of simple/primitive data types covering: integer, real, character, and string
- ◆ The use of no less than two different types of control structures
- ◆ The use of no less than two types of operators
- ◆ The implementation of at least one subroutine/procedure within the program
- ◆ Use of at least three comments embedded throughout the program

National Unit Specification: statement of standards (cont)

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

An assessor must endorse each candidate checklist with their name, signature and date.

The Assessment Support Pack (ASP) for this Unit provides sample assessment materials including assessor checklists, practical tasks and an instrument of assessment for the knowledge. Centres wishing to develop their own assessments should refer to the Assessment Support Pack to ensure a comparable standard.

National Unit Specification: support notes

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

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

Outcome 1

This Outcome covers software development principles and problem solving methods that are necessary in a programming environment.

The requirements needed in terms of hardware and software to enable the generation of different types of computer solutions.

In order to generate a computer program, the candidate will be made aware of the fact that such programs are usually developed through different stages (analysis, design, coding, implementation, testing, documentation, and maintenance). The stages should be clearly listed and described to enable a good exposure of the process of computer program generation.

The various programming techniques available currently will be highlighted focusing on advantages and disadvantages of each technique.

List the difference(s) between compiled programs that use languages such as C, C++, or VB to generate applications, and interpreted languages such as VB and scripts used for the Internet that need no more than an editor and a browser to run. The candidates should then be able to distinguish between the two types and extract both advantages and disadvantages of each.

Outcome 2

This Outcome covers identifying the simple/primitive data types such as numbers and characters and identifying modular techniques used when constructing computer programs, that is, the creation of subroutines otherwise known as procedures, functions, and/or methods.

Differences between input and output data types are described in this Outcome.

The various simple/primitive data types are listed with a brief description of each. Candidates will be shown the significance of each data type and the context of its usage.

A candidate must be made aware of the fact that computer programs are created in a modular way, in other words, using subroutines/procedures. Advantages of such methodology must be made clear to the candidates with simple illustrative examples. The examples do not have to be limited to programming but can also include real life scenarios.

Candidates must be shown how a computer program flows from the beginning to the end and how such a flow can be controlled. Therefore, different types of control structures are to be identified together with diagrams of the flow of control.

National Unit Specification: support notes (cont)

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

Outcome 3

This Outcome covers the definition and familiarisation of some of the common structured techniques used in formulating the logic of a software program - program design.

Description of how to initiate a program design by planning a solution using one or more structured programming design technique.

Description and illustration by examples of the various structured techniques used in designing software programs.

A candidate must learn how to implement a program design using pseudo-code and at least one graphical notation.

Outcome 4

This Outcome covers the implementation of a solution for a simple problem using a high-level programming language. The problem can be either provided by the centre or, alternatively, the candidate may suggest an appropriate scenario approved by the assessor.

Input data, in the form of simple/primitive type variables, will be used by the candidate in their programs. The variables will be declared and used properly throughout the candidate's program. The range of variables will include integers and real numbers, characters and/or strings¹.

A candidate will use different types of control structures, in the implemented solution.

The candidates will make use of different types of operators, that is arithmetic, assignment, and/or comparison (logical) operators in their implemented program solution.

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

Candidates will require access to appropriate computer hardware together with an appropriate programming environment. The type of environment needed will depend on the nature and type of programming language used. For example, a text editor and a browser is all that is needed for interpreted languages such as JavaScript. On the other hand, a compiler as well as an editor is needed to execute programs in a compiled structured language such as VB or Java. The choice of language will be entirely at the discretion of the centre.

¹ In object oriented languages strings are not considered as primitive variables

National Unit Specification: support notes (cont)

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

The overall ability of the candidates as well as the adopted teaching method(s) used will highly determine the amount of time needed to complete each Outcome. However, the following times can provide a rough guide:

Outcome 1	5 hours
Outcome 2	8 hours
Outcome 3	9 hours
Outcome 4	14 hours

The allocated timings allow for assessment and re-assessment where required.

OPPORTUNITIES FOR CORE SKILL DEVELOPMENT

In this Unit candidates are required to design solutions to given problems, this may provide an opportunity for developing aspects of the Core Skill in Problem Solving. In addition, some opportunities may arise for candidates to work together as part of the learning process. Candidates may also develop aspects of the Core Skill of Working with Others.

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

It may be appropriate for some of the evidence for this Unit to be produced using e-assessment. This may take the form of e-testing (for knowledge and understanding) and/or e-portfolios (for practical abilities). Centres using e-assessment must ensure the normal standards for validity and reliability are observed.

If a centre is presenting Outcome 1 and 2 of these assessments on-line the following assessment methods, where appropriate, may be selected:

Multiple choice
Drag and drop
Multiple response
Mix and match
Or a combination of the above

Outcomes 1 and 2 will be assessed by means of written and/or oral questions which will cover the knowledge and understanding.

Outcomes 3 and 4 will be assessed by means of practical exercises and an assignment brief.

An assessor observation checklist is required which authenticates that each candidate has completed all the Performance Criteria for Outcomes 3 and 4. An assessor must endorse each candidate checklist with their name, signature and date.

Documentary evidence in the form of code listing and screenshots of the generated output will be available to verify the candidate's completion of all practical tasks.

National Unit Specification: support notes (cont)

UNIT Computing: Programming in a High-level Language — Fundamentals (SCQF level 4)

The Assessment Support Pack (ASP) for this Unit provides sample assessment materials including assessor checklists, practical tasks and an instrument of assessment for the knowledge. Centres wishing to develop their own assessments should refer to the Assessment Support Pack to ensure a comparable standard.

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).