

## National Unit Specification: general information

**UNIT** Software Development (Intermediate 2)

**NUMBER** DF2Y 11

**COURSE** Computing (Intermediate 2)

### SUMMARY

This Unit is designed to develop knowledge and understanding of software development and to develop practical skills in software development through the use of a high level language within an appropriate software development environment. In particular, it will develop familiarity with standard language constructs in the context of short programs. It is not intended that candidates should develop complex programs involving sub-programs and combinations of constructs.

On completion of the Unit, the candidate should be able to apply this knowledge and understanding, and these skills to solve practical problems.

It is designed for candidates undertaking the Intermediate 2 Computing Course, but is also suitable for anyone wishing to develop an understanding of how software is developed using a high level programming language.

### OUTCOMES

1. Demonstrate knowledge and understanding of the principles of software development, software development languages and environments, high level language constructs and standard algorithms.
2. Demonstrate practical skills in the context of software development using contemporary hardware and an appropriate software development environment.

### 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:

- ◆ Intermediate 1 Computing Studies
- ◆ Standard Grade Computing Studies at General level

---

### Administrative Information

**Superclass:** CB

**Publication date:** April 2004

**Source:** Scottish Qualifications Authority

**Version:** 01

© Scottish Qualifications Authority 2004

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. The cost for each Unit specification is £2.50. (A handling charge of £1.95 will apply to all orders for priced items.)

## **National Unit Specification: general information (cont)**

**UNIT**                      Software Development (Intermediate 2)

### **CREDIT VALUE**

1 credit at Intermediate 2 (6 SCQF credit points at SCQF level 5\*).

*\*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**

There is no automatic certification of Core Skills or Core Skill components in this Unit.

## National Unit Specification: statement of standards

### UNIT Software Development (Intermediate 2)

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 principles of software development, software development languages and environments, high level language constructs and standard algorithms.

#### Performance Criteria

- a) Simple computing terminology is used appropriately.
- b) Simple descriptions and explanations are related to practical and familiar contexts.
- c) Descriptions of high level language constructs and standard algorithms are correct.
- d) Simple conclusions, predictions and generalisations are made from knowledge and understanding.

#### Evidence Requirements

Written or oral evidence that the candidate can describe and explain the principles of software development accurately and concisely. Evidence should be obtained using questions in a closed book test, under supervision, lasting no more than 45 minutes. The test must sample content (see Computing (Intermediate 2) Course content) in each of the following areas:

- ◆ the software development process
- ◆ software development languages and environments
- ◆ high level programming language constructs
- ◆ standard algorithms

(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 Software Development (Intermediate 2)

#### OUTCOME 2

Demonstrate practical skills in the context of software development using contemporary hardware and an appropriate software development environment.

#### Performance Criteria

- a) A range of appropriate hardware is used effectively.
- b) Common features of software are selected and used effectively.
- c) Practical tasks are planned and organised with detailed guidance.
- d) Practical tasks are undertaken in an appropriate range of simple contexts.

#### Evidence Requirements

Observation checklist showing that the candidate has carried out practical activities, demonstrating all of the following practical skills, as defined in the content statements (see Computing (Intermediate 2) Course content):

- ◆ analysis and design
- ◆ implementation of loops and conditional statements
- ◆ implementation of standard algorithm for input validation
- ◆ testing of software
- ◆ producing user and technical documentation
- ◆ evaluating software

Hard copy evidence should be provided of implementation and **one** other of these skills.

These practical skills may all be demonstrated in a single extended software development task, or in a number of smaller tasks.

The candidate will be allowed access to books, notes and online help while completing the task(s).

(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 Software Development (Intermediate 2)

This part of the Unit Specification is offered as guidance.

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

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

<b>Content Statement: software development process</b>
Description of the stages (in order) of the software development process: analysis, design, implementation, testing, documentation, evaluation, maintenance.
Description and exemplification of pseudocode and one graphical design notation (structure diagram or other suitable)
Description and exemplification of appropriate test data (normal, extreme and exceptional)
Description of the features of a user guide and a technical guide
Evaluation of software in terms of fitness for purpose, user interface and readability

<b>Content Statement: software development languages and environments</b>
Description and comparison of machine code and high level languages.
Explanation of the need for translation; description of the functioning of interpreters and compilers.
Description of the process of recording a macro and assigning it to a keystroke; description of a macro, and examples of the use of macros.
Description of the features and use of a text editor.

<b>Content Statement: high level programming language constructs</b>
Description and exemplification of the following constructs in an appropriate high level language: <ul style="list-style-type: none"><li>• Input and output, assignment, arithmetical operations (+,-,*,/,^ ) and logical operators (AND, OR, NOT).</li><li>• Fixed loops, conditional loops, simple and complex conditions, conditional statements (IF), nested loops.</li></ul>
Description and exemplification of numeric and string variables and 1-D arrays.
Description and exemplification of pre-defined functions.

## National Unit Specification: support notes (cont)

### UNIT Software Development (Intermediate 2)

Content Statement: Standard Algorithms
Description and exemplification of the following standard algorithm in an appropriate high level language: <ul style="list-style-type: none"><li>◆ input validation</li></ul>
Recognition of appropriate use of the following standard algorithms: <ul style="list-style-type: none"><li>◆ input validation</li><li>◆ find min/max</li><li>◆ count occurrences</li><li>◆ linear search</li></ul>

#### **GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT**

Candidates will require individual access to appropriate computer hardware and an appropriate software development environment throughout this Unit. While the learning will usually be achieved in the context of a single software development environment, candidates will benefit from having some experience of alternative software development environments.

The two Outcomes should be delivered in an integrated way rather than sequentially. For Outcome 2, the practical activities should be taught and used to illustrate and exemplify the knowledge and understanding required for Outcome 1.

The amount of time spent on each topic 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:

software development process	5 hours
languages and environments	5 hours
language constructs	20 hours
standard algorithms	6 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 a Course, the Course documentation will provide further information on teaching and learning in a Course context, including the identification of a number of 'themes' to facilitate holistic learning across the Course.

## National Unit Specification: support notes (cont)

### UNIT Software Development (Intermediate 2)

#### 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 examination conditions. 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 should be of a comparable standard.

Outcome 2 requires the candidate to demonstrate practical skills while developing software using an appropriate high level language environment. These practical skills may all be demonstrated in a single extended software development task, or in a number of smaller tasks. The skills will normally be demonstrated by the candidate during the teaching and learning activities of the Unit, rather than as separate formal assessment activities. The candidate will be allowed access to books, notes and online help while completing the task(s).

To gain success in this Outcome, the candidate must demonstrate practical skills in the following contexts and at an appropriate level as defined by the Content Statements (see Computing (Intermediate 2) Course content):

- ◆ analysis and design
- ◆ implementation of two language constructs
- ◆ implementation of the standard algorithm for input validation
- ◆ testing of software
- ◆ producing user and technical documentation
- ◆ evaluating software

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

Hard copy evidence is required to demonstrate implementation and one other of these skills; this need not be formal documentation – it could include hand-written design notes, hard copy of coding, or screen shots demonstrating implementation and/or testing.

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

#### SPECIAL NEEDS

This Unit Specification is intended to ensure that there are no artificial barriers to learning or assessment. Special needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments or considering special alternative Outcomes for Units. For information on these, please refer to the SQA document *Guidance on Special Assessment Arrangements* (SQA, September, 2003).