SCOTTISH
QUALIFICATIONS
AUTHORITY

COMPUTING
Higher

Fourth edition – published December 1999

SCOTTISH QUALIFICATIONS AUTHORITY

NOTE OF CHANGES TO ARRANGEMENTS FOURTH EDITION PUBLISHED ON CD ROM DECEMBER 1999

COURSE TITLE: Computing (Higher)

COURSE NUMBER: C017 12

National Course Specification

Course Details:	Core skills statements expanded

National Unit Specification:

All Units:	Core skills statements expanded

SCOTTISH QUALIFICATIONS AUTHORITY

National Course Specification

COMPUTING (HIGHER)

COURSE NUMBER C017 12

COURSE STRUCTURE

The course comprises two mandatory units and one optional unit:

Mandatory units:

D093 12	***Computer Systems (H)***	***1 credit (40 hours)***
D095 12	***Software Development (H)***	***1 credit (40 hours)***

Optional units - one selected from:

D101 12	***Artificial Intelligence (H)***	***1 credit (40 hours)***
D100 12	***Computer Programming (H)***	***1 credit (40 hours)***
D099 12	***Computer Networking (H)***	***1 credit (40 hours)***
D102 12	***Multimedia Technology (H)***	***1 credit (40 hours)***

All courses include 40 hours over and above the 120 hours for the component units. This may be used for induction, extending the range of learning and teaching approaches, support, consolidation, integration of learning, and preparation for external assessment.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Course Specification: course details (cont)

COURSE Computing (Higher)

RECOMMENDED ENTRY

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

- Computing Course at Intermediate 2 level
- Grade 1 or 2 in Standard Grade Computing Studies.

CORE SKILLS

This course gives automatic certification of the following:

Complete core skills for the course	Problem Solving	H
	IT	H
Additional core skills for the course	None	

For information about the automatic certification of core skills for any individual unit in this course, please refer to the general information section at the beginning of the unit.

Additional information about core skills is published in *Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Course Specification: course details (cont)

COURSE Computing (Higher)

RATIONALE

In recent years, computing has played an increasingly important role in modern society. The influence of computing systems has been pervasive, affecting work, home and leisure activities. Commercial and industrial practices have been greatly influenced by the availability of increasingly sophisticated computing systems at ever-decreasing cost. Developments such as the Internet have taken on new impetus and are of increasing importance to our society.

Higher Computing builds upon the knowledge and understanding of computing concepts considered at Standard Grade and Intermediate 2, providing a more focused study of the operation and organisation of computer systems combined with a more formal approach to the processes involved in the development of solutions to computing problems. This provides the opportunity for candidates with diverse computing backgrounds to consolidate and extend their experience and to prepare for further study.

Higher Computing provides an opportunity for candidates to gain an understanding of the underlying computing concepts and processes that drive information technology and to acquire skills in the development of computing solutions within a broad-based course, which reflects the wide range of computing. The course develops generic, transferable, practical competencies and an understanding of computing concepts that are applicable in a range of contexts and activities. Additionally, there is emphasis on the design, testing and evaluation of computing solutions.

The aims of this course are to:

- provide candidates with knowledge and understanding of underlying fundamental computing concepts, with regard to the operation and organisation of computer systems as a basis for the assimilation of future developments
- develop in candidates an appreciation of the applicability and potential of computing systems
- develop in candidates skills of analysis, synthesis, evaluation, communication and problem-solving within a computing context
- develop practical abilities in the use of computing technology
- provide intellectual stimulus and challenge, develop academic rigour and foster an enjoyment of the subject
- cater for career demands and personal developmental requirements across a range of candidate aspirations.

The knowledge and understanding gained in Higher Computing permit candidates to consider objectively the social, ethical and economic issues associated with the widespread personal and industrial use of computers. The further development of analytical and problem-solving abilities gained through the study of Higher Computing should enable candidates to have a creative role in a modern technological society.

National Course Specification: course details (cont)

COURSE Computing (Higher)

COURSE CONTENT

The pursuit of the aims stated above should lead to the achievement of learning outcomes dealing with knowledge and understanding, problem-solving and practical abilities.

The two mandatory units of the course provide the candidate with a balanced experience of the major aspects of the subject area. The third optional unit builds into the course the flexibility to be responsive to local needs and interests. By studying the component units within the context of the Higher Computing course, candidates' learning experience will be enhanced through having the opportunity to identify recurring themes, and through the development of practical and problem-solving skills that require the synthesis of knowledge and skills gained in the discrete units.

Undertaking the units as a coherent programme offers a number of benefits:

- together, the component units offer opportunities for delivery as a coherent, integrated, holistic experience
- balance and breadth of candidates' experiences and learning will be promoted
- both specific and core skills may be explored and developed
- skills and abilities developed through holistic and integrated activity support learning as a whole
- a candidate's abilities to sustain effort and concentration, come to conclusions, make decisions, complete a process and evaluate their work are developed.

While each unit has an appropriate mix of knowledge and skills represented within the outcomes, certain units may have more knowledge-based or practical outcomes than others. The proportion of outcomes relating to practical or cognitive competencies will reflect the domain of the unit. However, in the context of the course, there is an overall balance between practical and cognitive outcomes.

The unit specifications have been fully developed and provide detailed support notes to assist teachers and lecturers in their understanding of the outcomes and performance criteria. The support notes provide details about content *in the context of an individual unit*; this documentation provides details about content *in a course context*.

To ensure consistency of terminology, the meanings of the technical terms used throughout this documentation (including the unit specifications) were taken from the British Computer Society's publication entitled *A Glossary of Computing Terms* (published by Longman).

National Course Specification: course details (cont)

COURSE Computing (Higher)

UNIT CONTENTS

The detailed contents of each unit are defined within the respective unit specification and are not further described within this course documentation.

ASSESSMENT

To gain the award of the course, the candidate must achieve all the component units of the course as well as the external assessment. The external assessment will provide the basis for grading attainment in the course award.

When units are taken as component parts of a course, candidates will have the opportunity to demonstrate achievement beyond that required in attaining each of the unit outcomes. This attainment may, where appropriate, be recorded and used to contribute towards course estimates, and to provide evidence for appeals. Additional details are provided, where appropriate, with the assessment materials. Further information on the key principles of assessment is provided in the paper *Assessment* (HSDU, 1996) and in *Managing Assessment* (HSDU, 1998)

DETAILS OF THE INSTRUMENTS FOR INTERNAL ASSESSMENT

Each unit specification suggests ways of integrating assessment within units. By undertaking the units as part of a course, assessments can be integrated between units. The following table suggests one way of combining the assessment of unit outcomes within the Computing course at Higher level.

	Computer Systems				**Software Development**				**Computer Programming**				
Assessment	1	2	3	4	1	2	3	4	1	2	3	4	5
1	✓	✓	✓										
2		✓	✓										
3				✓									
4					✓	✓			✓	✓			
5							✓	✓			✓	✓	
6													✓

Table 7 – Assessment matrix for Computing at Higher level

It assumes that the optional unit is Computer Programming. The table shows how the 12 outcomes within the component units can be assessed using six assessment activities. Outline details of each instrument of assessment follow.

Assessment 1

This assessment would combine the assessment of the knowledge and understanding contained within Outcomes 1, 2 and 3 of the Computer Systems unit.

Assessment 2

This assessment would combine the practical elements of the Computer Systems unit into a single practical activity which would take the form of a practical assignment involving the candidate in using and organising a single-user computer system. Note that part of Outcomes 2 and 3 include practical activities as part of the performance criteria.

Assessment 3

This practical activity would involve candidates in selecting computer systems to match specific operational requirements.

Assessment 4

The Software Development (H) unit and the Computer Programming (H) unit are closely related and differentiated by outcome, performance criteria and/or range so there is considerable potential to integrate the assessment of these units. This proposed assessment would encompass Outcomes 1 and 2 from both units. The assessment would relate to the features of a contemporary software development environment.

Assessment 5

This assessment combines two outcomes from Software Development (H) with two outcomes from Computer Programming (H). These outcomes relate to the implementation stage of the software development process and require the candidate to use a range of software development tools.

Assessment 6

This outcome is not integrated with other outcomes and simply relates to the investigation of a contemporary issue relating to software development.

These outline descriptions give an indication of the type of assessment that could be used to integrate assessment and thereby reduce the assessment burden on candidates and staff. They are provided for illustration only. Fully developed assessment specifications will be produced as part of the National Assessment Bank.

National Course Specification: course details (cont)

COURSE Computing (Higher)

DETAILS OF THE INSTRUMENTS FOR EXTERNAL ASSESSMENT

External (course) assessment should provide opportunities to demonstrate:

- retention of knowledge, understanding and skills over a longer period of time
- the integration of knowledge, understanding and skills acquired in component units
- application of knowledge, understanding and skills in more complex contexts
- application of knowledge, understanding and skills in less familiar contexts.

The course assessment for Computing at Higher level will consist of two components with weightings as follows:

- coursework..... 30%
- written examination ..... 70%.

Coursework

The coursework component is designed to assess candidates' practical skills in applied computing across component units.

For each candidate, the teacher/lecturer should select existing evidence for two unit assessments from a list designated by SQA and from different units of the course. The evidence for each assessment should demonstrate the candidate's ability in the following four stages of applied computing – analysis, design, implementation and evaluation.

Currently, the designated list includes the following assessments from the Higher units:

UNIT	ASSESSMENT NUMBER	TITLE
Computer Systems	CS/04	Report for system specification
Software Development	SD/04	Report on problem solving
Artificial Intelligence	AI/03A	Report on problem solving using a declarative language
	AI/03B	Report on problem solving using an expert system
	AI/04	Report on investigation
Computer Programming	CP/04	Report on problem solving
	CP/05	Report on investigation
Computer Networking	CN/04	Report on investigation
Multimedia Technology	MMT/04	Report on investigation

National Course Specification: course details (cont)

COURSE Computing (Higher)

To establish the candidate's coursework grade, the teacher/lecturer should mark the selected assessments as follows:

1 Award 5 marks to each completed assessment if the evidence justifies a pass at unit level.
2 An assessment that was passed at unit level on first submission may be allocated up to 5 additional marks according to the criteria in the table below. These have been chosen to support a more holistic view of the candidate's applied computing skills.
3 If an assessment needed to be **re-submitted** at unit level, it should be awarded no additional marks.

CRITERIA	MARKS
High quality, clearly inter-related evidence for the four stages of applied computing ie analysis, design, implementation, and evaluation; and this evidence is tightly structured and displays a high level of subject expertise derived from the content of the unit.	5
Good quality, inter-related evidence for the four stages of applied computing ie analysis, design, implementation, and evaluation; and this evidence is well structured and displays a good level of subject expertise derived from the content of the unit.	3
Adequate quality, fairly well inter-related evidence for the four stages of applied computing ie analysis, design, implementation, and evaluation; and this evidence is reasonably well structured and displays an adequate level of subject expertise derived from the content of the unit.	1

4 Where the evidence for an assessment falls between these sets of criteria, the teacher/lecturer may award 4 or 2 marks as appropriate.
5 Adding the marks for pass at unit level, and the additional marks out of 5 for each task will give the candidate a mark out of 20.
6 The final mark allocated to the candidate for coursework is achieved by multiplying the mark out of 20 by 3 and dividing by 2. All half marks should be rounded up to the nearest integer.
7 Coursework marks from a centre will be subject to external moderation by SQA through sampling and/or visiting moderation. SQA will wish to assure itself that the allocation of marks across all candidates in a centre is appropriate.

Written examination

This will comprise a single paper of 2 hours and 30 minutes' duration. The examination will be set and marked by the Scottish Qualifications Authority.

The paper will be composed of 3 sections:

SECTION 1 (30% of the marks):
This will consist of questions requiring short responses and will be based on the two mandatory units. The questions will test both knowledge and understanding and problem solving. The ratio of knowledge and understanding to problem solving with be 2:1. The problems will be based in familiar contexts and be of a fairly straightforward nature. Candidates will be expected to tackle all questions.

SECTION 2 (40% of the marks):
This will take the form of questions which seek extended responses requiring structure and reasoning. The questions will involve problem solving and will be set in less familiar and more complex contexts than those in Section 1. The questions will be based on the two mandatory units. Some, but not all, will require the synthesis of knowledge across the two units. Candidates will have an element of choice in the questions that they will tackle.

SECTION 3 (30% of the marks):
This will have four sub-sections, one for each of the optional units. Candidates will be expected to undertake one of the sub-sections.

Each sub-section will test both knowledge and understanding and problem solving. The number of questions on knowledge and understanding will be small and the level of problem solving will equate to that in Section 2 above. Candidates will have an element of choice in each sub-section.

GRADE DESCRIPTIONS

Success in the course at Grades C, B and A will be determined by the sum of performances in both components (coursework and written examination). These performances will be assessed through detailed marking schemes and the resulting mark will be used to grade the candidate's overall performance.

The characteristics of candidates' performance at Grade C and at Grade A are described below. These descriptions will assist examiners in setting examination papers and will help internal assessors and moderators in determining a national standard when assessing candidates' coursework. The descriptions relate to the previously stated objectives of external assessment (see above).).

National Course Specification: course details (cont)

COURSE Computing (Higher)

GRADE C	GRADE A
• Retention of knowledge, understanding and skills over a longer period of time.	
Candidates are able to describe and explain *some* of the computing-related facts and concepts to the standard defined within the performance criteria.	Candidates are able to describe and explain *most* of the computing-related facts and concepts to the standard defined within the performance criteria *and exhibit higher cognitive competencies in relation to evaluation and selection.*
Candidates are able to apply their skills in the context of an integrated assessment that encompasses outcomes from more than one unit to the standards defined by the associated performance criteria.	Candidates are able to apply their skills in the context of an integrated assessment that encompasses outcomes from more than one unit to standards that *exceed* the associated performance criteria *and exhibit higher order skills in relation to problem resolution and independent working.*
• The integration of knowledge, understanding and skills acquired in component units.	
Candidates are able to apply their knowledge and understanding in the context of specific units.	Candidates are able to integrate knowledge and understanding acquired *in different component units and exhibit higher cognitive competencies in relation to selection and evaluation* .
Candidates are able to apply their skills in the context of a practical coursework which integrates outcomes from more than one unit.	Candidates are able to apply their skills in the context of a practical coursework which integrates outcomes from more than one unit *and skills are comfortably transferred and synthesised.*

National Course Specification: course details (cont)

COURSE Computing (Higher)

GRADE C	GRADE A
• Application of knowledge, understanding and skills in more complex contexts.	
Candidates are able to apply their knowledge and understanding in straightforward contexts, directly related to the component units.	Candidates are able to apply their knowledge and understanding in contexts that derive from outcomes *in more than one unit*
The artifact that results from practical activity meets the standards defined by the associated performance criteria.	The artifact that results from practical activity *exceeds* the standards defined by the associated performance criteria.
• Application of knowledge, understanding and skills in less familiar contexts.	
Candidates are able to apply their knowledge and understanding in familiar contexts.	Candidates are able to apply their knowledge and understanding in *less familiar* contexts *and exhibit skills in transferring skills from a familiar environment to a less familiar environment.*
Candidates are able to carry out defined tasks to the standards defined in the associated performance criteria.	Candidates are able to resolve *non-routine problems* that arise during their practical activity and *exhibit skills in relation to problem solving and independent working.*

Table 1 – Grade descriptions for Computing at Higher

National Course Specification: course details (cont)

COURSE Computing (Higher)

RELATIONSHIP OF THE COURSE TO THE COMPONENT UNITS

The course consists of three component units, plus an additional 40 hours of study. An external examination tests the candidates' knowledge and understanding of the content covered in all three units and their ability to demonstrate and to integrate skills acquired throughout the course.

The criteria for a Grade C in this course closely reflect the level of competence required for success in the component units. However, the external assessment of the course makes specific additional demands on the candidates.

The external assessment grade descriptions require that candidates demonstrate the ability to integrate the competences demonstrated in the component units of the course. The knowledge and understanding, skills and ability to use concepts that have been acquired during the units must be retained and demonstrated in the answering of unseen questions.

APPROACHES TO LEARNING AND TEACHING

Advice on the delivery of each component unit is given within the support notes section of each unit specification. A Subject Guide will provide generic advice applicable to all computing-related courses and units. This section provides further guidance on teaching and learning in a course context. In addition to the unit and course documentation and the Subject Guide, support materials will be provided to assist teachers and lecturers in their delivery of courses and units.

In designing the Computing courses, a number of topics were identified as important at various levels within the framework of provision. These multi-level topics (or themes) relate to the following areas:

- the principles of computer systems
- practical abilities in using computer systems
- applications of computer systems
- software development process
- problem-solving skills
- objects and operations
- stimulus to development and contemporary trends
- social, legal and ethical implications.

Some, or all, of these topics appear at every level. For example, the principles of computer organisation arise at Access, Intermediate 1, Intermediate 2, Higher and Advanced Higher (to a greater or lesser extent); software development is a part of every course from Intermediate 1 to Advanced Higher level; and contemporary trends are investigated within various units at various levels.

An important consideration is to ensure that the treatment of any one of these topics is coherent and a *progression matrix* was produced for the computer systems, software development and computer applications units. These matrices are provided in the Subject Guide. They define the contents of these topics at each level and ensure that candidates working through the framework have a progressive experience. They are also useful to teachers and lecturers for explaining the content of a unit at a given level and units should be considered in the context of these tables.

The identification of these themes permits the teacher/lecturer to take a more holistic approach to teaching and learning so that knowledge and skills gained as part of one unit can contribute to another (at the same level or another level). The following table illustrates how these topics are distributed between units within the Higher course.

Integrative topic	Computer Systems	Software Dev.	AI	Computer Prog.	Computer Net-working	Multi-media
Principles	✓		✓		✓	
Practical abilities	✓	✓	✓	✓	✓	✓
Applications	✓		✓		✓	✓
Software development		✓		✓		
Problem-solving		✓	✓	✓		✓
Object and operations	✓	✓	✓	✓		✓
Developments			✓	✓	✓	✓
Implications	✓		✓		✓	

Table 2 – Integration of learning and teaching within Computing at Higher level

The above table illustrates that practical skills are a part of every unit within this course and teachers/lecturers can help candidates acquire these skills in a holistic manner rather than teaching these skills on an outcome-by-outcome basis. So, for example, when candidates acquire practical skills as part of the Computer Systems unit, the knowledge and skills gained can also contribute to other units within the course. The identification of these integrative topics will help teachers/lecturers deliver the necessary knowledge and skills in a holistic and 'natural' manner.

The content within each unit is described in terms of the computing concepts which are to be studied. It is important that these concepts are illustrated through choice of suitable contexts. These should be chosen to reflect an overall balance between small and large-scale uses of computers and should be representative of commercial, industrial, educational, scientific, domestic and leisure fields.

It is important that candidates do not consider the content of the component units in isolation. An important aspect of this course is the ability to draw comparisons and identify similarities between concepts studied in component units, and to combine knowledge and skills gained in more than one unit and apply them to a range of contexts.

A variety of support materials will be produced to assist teachers and lecturers in their delivery of the component units and assessment materials will be provided as part of a National Assessment Bank. A wide range of learning media is available to assist teachers/lecturers in delivering computing-related topics. These materials are available in a variety of formats (such as videotape and CD ROM). The Internet is a rich source of teaching and learning materials for Computing courses.

Each unit specification has a section on learning and teaching and this, together with the above advice on learning and teaching in a course context, should assist teachers and lecturers in delivering the component units. Further details on learning and teaching will be provided in the Subject Guide which will give additional information on the delivery of Computing courses and units.

40 hours of flexible time

The course allows 40 hours of additional flexible time. Appropriate activities that could be undertaken might include:

- preparing for external assessment
- formative assessment
- consolidation of learning
- remediation and re-assessment
- extending the range of study
- developing study skills
- visits.

SPECIAL NEEDS

This course 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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).

SUBJECT GUIDES

A Subject Guide to accompany the Arrangements Documents has been produced by the Higher Still Development Unit (HSDU) in partnership with the Scottish Consultative Council on the Curriculum (SCCC) and Scottish Further Education Unit (SFEU). The Guide provides further advice and information about:

- support materials for each course
- learning and teaching approaches in addition to the information provided in the Arrangements document
- assessment
- ensuring appropriate access for candidates with special educational needs.

The Subject Guide is intended to support the information contained in the Arrangements document. The SQA Arrangements documents contain the standards against which candidates are assessed.

SCOTTISH QUALIFICATIONS AUTHORITY

National Unit Specification: general information

UNIT Computer Systems (Higher)

NUMBER D093 12

COURSE Computing (Higher)

SUMMARY

This unit is designed to develop knowledge and understanding of the principles of computer systems and provide an opportunity to apply this knowledge through the use of contemporary hardware and software.

OUTCOMES

1 Demonstrate knowledge and understanding of the principles of computer organisation.
2 Demonstrate knowledge and understanding of computer software.
3 Demonstrate knowledge and understanding of the input, storage and output of data.
4 Select a computer system to match specific operational requirements.

RECOMMENDED ENTRY

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

- the corresponding unit at Intermediate 2
- Intermediate 2 Computing
- Intermediate 2 Information Systems
- Grade 1 or 2 at Standard Grade in Computing Studies.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Unit Specification: general information (cont)

UNIT Computer Systems (Higher)

CREDIT VALUE

1 credit at Higher.

CORE SKILLS

There is no automatic certification of core skills or core skills components in this unit.

Additional information about core skills is published *in Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Unit Specification: statement of standards

UNIT Computer Systems (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 principles of computer organisation.

Performance criteria

(a) Explanation of the principles of computer organisation is accurate and concise.
(b) Description of methods of data representation is technically accurate
(c) Description of characteristics of different computer systems is technically accurate.
(d) Description of the factors that affect system performance is accurate and concise.

Note on range for the outcome

Principles: two-state machine; structure of CPU; memory; stored program concept; fetch-execute cycle; interfaces.
Data representation: text; integer; real; bit-mapped graphic; vector graphic.
Computer systems: desktop; portable; networked; mainframe.
Characteristics: processor; peripherals; resource sharing; topology
Factors: bus width; clock speed; memory; buffers; data integrity.

Evidence requirements

Written or oral evidence that the candidate can explain the principles of computer organisation as detailed in PCs (a) to (d) across all classes in the range. In describing data representation, the candidate should exemplify the representation of each type of data for two systems with a different data word length.

National Unit Specification: statement of standards (cont)

UNIT Computer Systems (Higher)

OUTCOME 2

Demonstrate knowledge and understanding of computer software.

Performance criteria

(a) Description of the types and purposes of system software is correct.
(b) Description of the types and purposes of programming languages is correct.
(c) Description of the system requirements for application software is technically correct.
(d) Use of computer software to set up and customise a computer system is efficient and effective.

Note on range for the outcome

System software: single-user operating systems; network operating systems; utility software; compiler; interpreter.
Types of programming language: procedural; declarative; event-driven; scripting.
Purposes of programming language: general-purpose; scientific; commercial; artificial intelligence; system.
Types of application software: text; number; data; communications (including Internet); graphics; multimedia (including sound and video); integrated.
Computer software: operating system; utility software; application software.

Evidence requirements

Written or oral evidence that the candidate can describe the software characteristics of different computer systems as detailed in PCs (a) to (c) for all classes in the range.

Performance evidence that the candidate can use computer software to set up and customise a computer system as detailed in PC (d). For the purposes of core skill certification, the candidate must use virus checking software.

OUTCOME 3

Demonstrate knowledge and understanding of the input, storage and output of data.

Performance criteria

(a) The characteristics of input, storage and output devices are correctly explained.
(b) The comparison of the characteristics of contemporary products is accurate.
(c) Organisation of backing storage is efficient and data backup is systematic and secure.
(d) Retrieval and storage of data matching compound search criteria are performed efficiently and responsibly.

Evidence requirements

Written or oral evidence that the candidate can explain the characteristics of input, storage and output devices as detailed in PC (a). The explanations must relate to at least two input devices selected from modified keyboards, digitisers, sound, video, digital cameras; at least two storage devices selected from magnetic, optical, magneto-optical, DAT tape; and at least two output devices selected from sound, video, printer, colour printer, film, multiscan monitors.

Written or oral evidence that the candidate can compare the characteristics of contemporary products as detailed in PC (b) for all classes in the range. At least two input devices and two output devices and two storage devices must be compared with respect to these characteristics.

For the purposes of explanation and comparison the characteristics must include cost, and as appropriate, access, accuracy, capacity and speed.

Performance evidence that the candidate can use system software to organise backing storage efficiently as detailed in PC (c). The candidate must provide evidence of use of data backup and virus protection software.

Performance evidence that the candidate can retrieve and store data that matches compound search criteria (involving the use of Boolean operators) as detailed in PC (d). The evidence should demonstrate that the candidate has adhered to legal and health requirements. For the purposes of core skill certification, at least two different sources of information should be searched.

OUTCOME 4

Select a computer system to match specific operational requirements.

Performance criteria

(a) Analysis of the operational requirements correctly identifies the nature of the problem and the main features relevant to the solution are clearly described.
(b) System specification is clear and complete, and satisfies the operational requirements.
(c) The system specification is justified in terms of functionality, compatibility and cost.

Note on range for the outcome

System specification: computer hardware; computer software.

Evidence requirements

Written or oral evidence that the candidate can select a computer system to match specific operational requirements as detailed in PCs (a) to (c) for all classes in the range. Evidence will be in the form of a system specification together with a report which provides justification of the selection.

National Unit Specification: support notes

UNIT Computer Systems (Higher)

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

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

GUIDANCE ON CONTENT AND CONTEXT FOR THIS UNIT

This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Higher level. The details of content given earlier provide information on the delivery and assessment of this unit within a course context.

Corresponding to Outcomes 1–4

This unit is the fourth in a series of units relating to Computer Systems and other units are available at Access, Intermediate 1, Intermediate 2 and Advanced Higher levels. As such, candidates should possess knowledge and skills in computer systems prior to undertaking this unit or be capable of accelerated learning in order to accomplish the outcomes in the recommended time.

Reference may be made to specific Internet resources within this unit specification; whenever possible a related URL has been supplied.

Outcome 1

This outcome relates to the principles of computer systems. In relation to Von Neumann architecture, the candidate should be able to:

- describe the organisation of a computer as a two-state (binary) machine
- explain the organisation of a computer to include CPU, memory, storage, I/O devices, buses (address data and control)
- explain the organisation of a computer memory to include the concept of addressability
- explain the organisation of the CPU as ALU, control unit and registers - the details of what each of these registers is for is NOT required
- describe the stored program concept including the idea of a fetch-execute cycle
- explain the need for interfaces between the CPU and peripheral devices.

Candidates will be expected have a knowledge of binary to allow them to appreciate the internal representation of data. Candidates should be able to describe this internal representation for:

- graphics: bit-mapped and vector in terms of advantages and disadvantages
- text: character set and the common standard (ASCII)
- number: integer - accuracy and size; floating point - implications of mantissa and exponent for range and accuracy.

National Unit Specification: support notes (cont)

UNIT Computer Systems (Higher)

At this level the candidate should be able to describe and distinguish between the hardware characteristics in terms of input, process, storage and output devices for single-user (desktop, laptop, palmtop), multi-user (mainframe) and networked systems. Additionally, the candidate should be able to describe networked systems in terms of their topology.

The candidate should be able to explain that different factors can have an impact on system performance. These factors will include bus width and processor clock speed.

Outcome 2

This outcome concerns the different types of software involved in computer systems. The candidate should be able to describe the main functions of an operating system and to be aware that one model of an operating system considers it to be composed of several layers (although the specific nature of each layer need not be known). The candidate should be able to describe the additional features of a network operating system with particular reference to multi-user access, security, file and print services and data sharing.

The candidate should be able to describe the types of programming languages suitable for education (including teaching programming), science, communications, data processing, artificial intelligence, Internetworking and real-time systems. The candidate should be aware of the different features of compilers and interpreters.

The candidate should develop awareness of commercially available software and the de facto standards which are used by the software. These standards could include standards for graphics, video, audio and communication. The candidate should be able to identify contemporary software and typical system requirements for document processing, numerical processing, data storage and retrieval, multimedia and Internet access. The candidate should be encouraged to research contemporary configurations for these applications (using internal and external sources of information such as textbooks and magazines) based on actual hardware and software products. Only the major components of each system should be identified (such as type of system, peripherals, system and application software); there is no requirement to identify low-level characteristics of the system such as word size or clock speed.

This outcome also relates to the candidate's use of computer systems. The candidate should be able to set up and customise a system for efficient use for a particular application At this level candidates would be expected to change hardware and/or software parameters to make efficient and effective use of the computer system. For example, candidates might experiment with threshold settings while scanning a graphic or may choose the resolution of a printer or may increase the memory allocated to an application. These can be used to exemplify the control the user has over a system. In some cases system parameters may be changed but clearly this should only be carried out where it is safe. Centres will make their own decision about which parameters can be changed, depending on the hardware/software platform available and their technical support resources. For the purposes of core skill certification, candidates must use virus checking software on the customised system.

National Unit Specification: support notes(cont)

UNIT Computer Systems (Higher)

Outcome 3

This outcome relates to the characteristics of input and output devices in contemporary use, contemporary storage devices and the organisation of storage. Candidates should be able to make comparisons between the products from different manufacturers for each type of input device. The criteria for each comparison will be those appropriate to the input device. The use of the word contemporary is deliberate and will require candidates to access current product information. For many manufacturers this data can be obtained from the company Web site. This will therefore involve the candidates in accessing data from a remote source and capturing it to local storage. A similar approach should be taken for the output devices and storage devices.

In discussing the input and output of sound and video, the candidate should familiarise himself/herself with the standards for data compression associated with the input and output of such data. The resolution of scanners and digitisers should also be covered

The candidate should be aware that there are different standards available in backing storage and be able to describe the general characteristics of these standards. The same knowledge should be demonstrated for output devices. Candidates should be encouraged to consider the most appropriate criteria for the discussion of system components, for example printing devices might be considered in terms of cost, dots per inch and pages per minute.

Candidates would be expected to organise backing storage efficiently using a hierarchical filing system, giving due regard to the balance between breadth and depth and appropriate names for files and folders (directories). Candidates should make regular backups of their own files and frequently check their files for viruses. The candidate should be able to search a remote source for data and download this data to local storage. For the purposes of core skill certification, at least two sources must be used, one of which may be local (such as CD ROM). The candidate should be aware of the wider implications of using systems. This will include health and safety issues related to the design and use of computers as well as the financial and legal implications of obtaining data from these sources.

Outcome 4

This outcome relates to the selection of computer systems to match specific applications. Suitable applications include office applications, financial applications, entertainment, publishing applications, educational uses, research and communications.

Candidates should be able to analyse an operational requirement specification and communicate clearly the critical system requirements and the boundaries of the problem.

Candidates are then required to select hardware and software (including system and application software). It is anticipated that candidates will use a variety of internal and external sources of information to identify suitable systems. It is likely that a case study may form a suitable context for this outcome. The candidates should be able to describe the resulting system in terms of hardware and software.

Candidates must be able to evaluate and justify the solution and assess critically its adequacy in terms of functionality, compatibility and cost, and be able to suggest possible enhancements. So, for example, selecting incompatible hardware and software is not acceptable; similarly, the hardware and software must be capable of carrying out the required tasks at an acceptable speed; and the cost of the system must be within a defined budget.

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

This is the fourth in a series of units relating to Computer Systems. The content/context section (above) provides additional guidance on the delivery of the outcomes. This section focuses on learning and teaching.

Candidates will require access to appropriate computer hardware and software at various stages within this unit. To enliven learning, the use of video, audio and multimedia learning aids is recommended. While the distribution of time between the outcomes will vary, candidates may be expected to complete each outcome within the following time scale:

Outcome 1 ..... 8 hours
Outcome 2 ..... 12 hours
Outcome 3 ..... 12 hours
Outcome 4 ..... 8 hours

If this unit is delivered as part of a course (or programme of units), then the course documentation will provide further information on teaching and learning in a course context. This documentation will identify a number of ‘themes’ to facilitate holistic learning and teaching across the course.

National Unit Specification: support notes(cont)

UNIT Computer Systems (Higher)

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

Centres may use the instruments of assessment which are considered to be most appropriate. Examples of instruments of assessment which could be used are as follows:

Outcome 1	Extended-response questions on the principles of computer organisation.
Outcome 2	Extended-response questions on the types and purposes of computer software.
Outcome 3	Extended-response questions on the characteristics of peripheral devices.
Outcome 4	Practical assignment involving the selection of a computer system to match operational requirements.
Outcome 5	Practical assignment involving the set-up, customisation and organisation of a microcomputer system.

During the work of the unit, candidates should have several opportunities to develop their practical skills and should be assessed at appropriate points. Terminology should be presented in context throughout the module. Where the candidate is unsuccessful in achieving an outcome, provision should be made for remediation and re-assessment.

Written evidence may take various forms including hand-writing and word-processed text or other forms of written communication that are more suited to candidates with physical disabilities. Candidates should not be required to produce a specific form of written text (such as word-processed text) unless this is stipulated within the unit specification or instrument of assessment

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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).

SCOTTISH QUALIFICATIONS AUTHORITY

National Unit Specification: general information

UNIT Software Development (Higher)

NUMBER D095 12

COURSE Computing (Higher)

SUMMARY

This unit is designed to extend knowledge and understanding of the software development process and of a software development environment. It provides the opportunity to implement features of a selected software development environment, to use its facilities and to apply a systematic methodology to produce a software solution to a problem using an appropriate software development environment.

OUTCOMES

1 Explain aspects of the software development process.
2 Explain features of a software development environment.
3 Implement features and use facilities of a software development environment.
4 Produce a solution to a problem using a 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 experience):

- Software Development (Int 2) unit
- Computing course at Intermediate 2 level
- Grade 1 or 2 at Standard Grade in Computing Studies.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Unit Specification: general information (cont)

UNIT Software Development (Higher)

CREDIT VALUE

1 credit at Higher.

CORE SKILLS

This unit gives automatic certification of the following:

Complete core skills for the unit	Problem Solving H
Additional core skills for the unit	None

Additional information about core skills is published *in Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Unit Specification: statement of standards

UNIT Software Development (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

Explain aspects of the software development process.

Performance criteria

(a) Stages and iterative nature of the software development process are explained accurately and concisely.
(b) Tools and techniques of the software development process are explained accurately.
(c) Desirable characteristics of software are explained accurately.
(d) Algorithms are designed and described accurately.

Note on range for the outcome

Stages: analysis; design; implementation; testing; documentation; evaluation; maintenance.
Tools and techniques: design methodology; test data; structured listing; error reporting; module libraries.
Characteristics: robustness; reliability; portability.
Algorithms: input validation; linear search; counting occurrences; finding maximum and minimum.

Evidence requirements

Written or oral evidence that the candidate can explain aspects of the software development process as detailed in PCs (a) to (c) for all appropriate classes in the range.

Written or diagrammatic evidence that the candidate can design and describe the algorithms detailed in PC (d).

National Unit Specification: statement of standards (cont)

UNIT Software Development (Higher)

OUTCOME 2

Explain features of a software development environment.

Performance criteria

(a) Features of a software development environment are defined correctly.
(b) Purposes of features are explained correctly.
(c) Means of implementing features are explained correctly.

Note on range for the outcome

Features: control; data storage; data flow.

Evidence requirements

Written or oral evidence that the candidate can explain features of software development environments as detailed in PCs (a) to (c) for all classes in the range. The explanation should include at least one from each of the classes: control, data storage and data flow. For this unit, the classes cover: complex conditionals, local variables, global variables, parameter passing by value, parameter passing by reference, one-dimensional arrays.

OUTCOME 3

Implement features and use facilities of a software development environment.

Performance criteria

(a) Syntax and semantics of features are used correctly.
(b) Implementations of features are correct and complete.
(c) Algorithms are implemented correctly.
(d) Characteristics which allow for maintenance of implementations are used appropriately.
(e) The environment's facilities are used effectively and efficiently.

Note on range for the outcome

Features: modularity; control; data storage; data flow; operations.
Facilities: editor; translator; error tracing.

Evidence requirements

Performance evidence that the candidate can implement features and use facilities of a software development environment as detailed in PCs (a) to (e) for all classes in the range. Evidence should cover use of complex conditionals, logical operations, local and global variables, and passing parameters by value and by reference. Evidence of implementation of algorithms should include an input validation algorithm plus one other from linear search, counting occurrences, maximum and minimum.

National Unit Specification: statement of standards (cont)

UNIT Software Development (Higher)

OUTCOME 4

Produce a solution to a problem using a software development environment.

Performance criteria

(a) The stages of the software development process are applied correctly.
(b) A working solution to a problem is produced.
(c) The solution is evaluated against given criteria.
(d) The results of the software development process are communicated clearly and concisely.

Note on range for the outcome

Stages: analysis; design; implementation; testing; documentation.
Criteria: user interface; fitness for purpose; maintainability; robustness; reliability; portability.

Evidence requirements

Performance evidence that the candidate can produce a solution to a problem using a software development environment as detailed in PCs (a) to (b) for the relevant classes in the range. The solution should include both user and technical documentation.

Written or oral evidence that the candidate can evaluate and report on a solution as detailed in PCs (c) to (d) for relevant classes in the range.

National Unit Specification: support notes

UNIT Software Development (Higher)

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

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

GUIDANCE ON CONTENT AND CONTEXT FOR THIS UNIT

This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Higher level. The details of content given earlier provide information on the delivery and assessment of this unit within a course context.

Corresponding to Outcomes 1–4

This unit is the third in a series of units relating to Software Development and other units are available at Intermediate 1, Intermediate 2 and Advanced Higher levels. It should be noted that Software Development (H) is a core unit which builds on learning in the core of Intermediate 2. As such, it is intended to consolidate earlier learning in this area and to lay a foundation for further study of the topic.

Progression through this series of Software Development units is shown in the relevant progression matrix (see Subject Guide). It should be noted that four main themes provide the structure for progression. Of these themes, two are associated with knowledge and understanding of computing facts and concepts and two with practical skills and abilities. The former themes are: software development process (stages; techniques and tools; characteristics; algorithms); and software development environment features. The latter themes are: working in a software development environment (implementation of features, implementation of algorithms, implementation of maintenance characteristics, use of environment's facilities); and problem solving (analysis, design, implementation, testing, documentation, evaluation).

Progression through the Software Development units is promoted in a number of ways. Introduction of new aspects increases breadth of study through units. Where the same aspects are included in units, depth of study is the key to progression. This depth can be manifested in different ways, for example: the level of cognitive skill becomes higher (eg recall, describe, explain); the nature of response is increasingly demanding (eg short-to-extended response); the level of difficulty of questions/problems (eg simple/complex, familiar/unfamiliar contexts) increases. In all units, candidates are expected to use correctly and in context terminology appropriate to the level of study. For example, within the context of software development, the term 'user interface' refers to factors which relate to the interaction between the user and the developed software solution. Where appropriate, guidance is given about the appropriate level of treatment of terms within the unit.

When studying a core course unit at a particular level, it is assumed that candidates are able to use knowledge and skills which are included in mandatory units (but not optional units) in the series at lower levels. Study of an optional unit assumes knowledge and skills in the mandatory unit at the same level. In the case of Software Development (H), prior knowledge and skills is assumed from Software Development (Int 2), although it is not prerequisite that a candidate has studied the unit.

The term *'software development environment'* refers to any environment whose features and facilities can be used to produce solutions to problems. An integrated environment with a 'programming' language (using the term in its very broadest sense) and facilities such as text editor and debuggers would be ideal. An imperative language, a declarative language, an applications package with a high functional capability, an object-oriented language, a scripting language or an authoring environment are all suitable. The features contained in the statement of standards are intended to be sufficiently open to facilitate delivery of the Software Development units in this wide range of environments. The support notes are exemplified in terms of a procedural high-level language. Implementations in other environments should be broadly compatible.

At Higher level, in-depth knowledge and skills should be developed within the context of one selected software development environment. Whenever possible, learning should be enhanced through comparisons with other environments. For example, if the main emphasis of study is in a high-level language environment then study of a higher-level language within a general-purpose package or a 4GL would be desirable. However, it should be noted that assessment takes place with respect to only one software development environment.

Outcome 1

In this outcome, candidates are required to explain aspects of the software development process. In completing this outcome, candidates will explain accurately and concisely the stages and iterative nature of the software development process (PC (a)) and explain accurately tools and techniques of the software development process (PC (b)). In addition, desirable characteristics of the software development process are to be explained accurately (PC (c)). The candidate is also required to design and describe algorithms accurately using an appropriate medium (PC (d)). Each performance criterion is now described in turn.

a) Stages and iterative nature of the software development process are described accurately and concisely.

The stages contained in the range are:

- analysis
- design
- implementation
- testing
- documentation
- evaluation
- maintenance.

Candidates should be able to give an accurate and concise description of each stage of the software development process:

- analysis relates to the understanding of the problem and the conversion of a rough outline of the problem into a precise problem specification which may include problem inputs and outputs
- design involves the methodical planning of the solution to the problem using a recognised design methodology, for example top-down design using structure diagrams or pseudo-code
- implementation requires the translation of the design plans into instructions that the computer can understand and the production of internal documentation
- the documentation includes a user guide and a technical guide
- evaluation involves reviewing the solution against suitable criteria
- maintenance involves making changes in the form of corrections or updates to the program at some time in the future and must be facilitated by aspects of good practice in software development.

b) Tools and techniques of the software development process are explained accurately.

The tools and techniques contained in the range are:

- design methodology
- test data
- structured listing
- error reporting
- module libraries.

The candidate should be able to give accurate explanations of the tools and techniques, including reasons for their use:

- two design methodologies - for example, a structure diagram and textual representation using pseudo-code of a stepwise refinement technique
- test data - to check that software works properly, candidates would be expected to have a knowledge of systematic and comprehensive testing
- structured listing - a formatted display or printout of solution code
- error reporting - the generation within the environment of messages indicating errors
- module libraries - pre-defined routines to support and speed up the software development process, part of the hierarchy of software.

c) Desirable characteristics of software are explained accurately.

The desirable characteristics contained in the range are:

- robustness
- reliability
- portability.

The candidate should be able to explain accurately:

- robustness - ability of software to cope with errors during execution without failing
- reliability - how well software operates without stopping due to design faults
- portability - the ability of software to be used on a computer other than that for which it was designed.

d) Algorithms are designed and described accurately.

The algorithms contained in the range are:

- input validation
- linear search
- counting occurrences
- finding maximum and minimum.

The candidate should be able to design and describe each of the listed algorithms. It is emphasised that there is no requirement to implement all of these.

Possible activities might include the use of small scale case studies to exemplify the stages of software development and its iterative nature. Good use can be made of prepared examples to illustrate the characteristics of robustness, reliability and portability. Much can be gained from the use of bad examples as well as good examples. These examples should, whenever possible and appropriate, cover different software development environments. To enhance their understanding, candidates should be encouraged through activities to develop confidence in using a design methodology, constructing complete sets of test data and utilising the error-tracing facilities of the chosen software development environment. The use of module libraries should be encouraged as a means of speeding up software development. Illustrative examples of algorithms in action would provide a useful basis on which to develop knowledge and understanding.

Outcome 2

This outcome relates to knowledge and understanding of features of a software development environment. Candidates are expected to define correctly features of a software development environment (PC (a)), explain correctly the purposes of these features (PC (b)) and explain correctly how the features are implemented (PC (c)) in a selected environment. Each performance criterion is now described in turn.

a) Features of a software development environment are defined correctly.

The features listed in the range are:

- control
- data storage
- data flow.

At this level, the candidate must be able to give comprehensive definitions of features.

For example, in a procedural language, the candidate should be able to define:

- selection (control) - complex conditions and decisions
- data storage - local and global variables; one-dimensional arrays
- data flow - passing parameters by value and by reference.

Note that, when using other environments, corresponding aspects of features should be defined.

a) Purposes of features are explained correctly.

The features listed in PC (a) apply also in this context. The candidate should be able to explain in some detail the purposes of features, with illustrative examples of when their use is appropriate.

b) Means of implementing features are explained correctly.

The features listed in PC (a) apply also in this context. The candidate should be able to explain the means of implementing features in one software development environment. For example, in a procedural language, with respect to selection and multiple conditions, the candidate should be able to explain the uses CASE and nested IF constructs.

This outcome provides candidates with the opportunity to gain an overview of extended features of a software development environment. Content is likely to be delivered alongside that of Outcome 3 with knowledge and understanding being reinforced by practical activity. Candidates could be introduced to new features one at a time or in small related groups, then carry out simple practical exercises, then answer questions related to PCs (a) to (c). Practical exercises might involve modification of given implementations or implementation of relevant features from scratch. It would be beneficial to learning if candidates can compare implementations of features in different software development environments.

Outcome 3

This outcome relates to practical skills and abilities associated with use of fundamental features and facilities of a software development environment and the implementation of algorithms. Candidates are expected to use syntax and semantics correctly (PC (a)) when producing working implementations of particular features (PC (b)) and selected algorithms (PC (c)). All implementations should be maintainable (PC (d)). When carrying out practical exercises, candidates should use effectively and efficiently editing and error-tracing facilities of the software development environment (PC (e)). Each performance criterion is now described in turn.

a) Syntax and semantics of features are used correctly.

Features included in the range are:

- modularity
- control
- operations.
- data storage
- data flow

At this level, the candidate must be able to use correct syntax and semantics for the range features in one software development environment. For example, in a procedural language, the candidate is expected to use correct syntax and semantics for:

- modularity - construction of functions
- selection (control) - complex conditions and decisions
- operations - logical operators
- data storage - local and global variables; one-dimensional arrays
- data flow - passing parameters by value and by reference.

Note that, when using other environments, corresponding aspects of features should be identified.

b) Working implementations of features are produced.

The features listed in PC (a) apply also in this context. As well as using correct syntax and semantics, candidates are expected to be able to complete short practical exercises which demonstrate that they can use correctly the range features in one software development environment. For example, in a procedural language, the candidate is expected to use correctly features detailed above. Whenever possible, practical exercises should cover several features or aspects of the features.

c) Algorithms are implemented correctly.

There is no range statement associated with this performance criterion. In the selected software development environment, candidates must implement at least two algorithms, including:

- input validation

and one other from:

- linear search
- count occurrences
- finding maximum and minimum.

The implementation may be based on an algorithm designed and described by the candidate or on a design provided by the teacher or lecturer.

d) Characteristics which allow for maintenance of implementations are used appropriately.

Candidates are expected to be able to implement features in a way which facilitates correction and updating in future. For example, in the context of a procedural language, candidates should produce programs which have a modular structure and which are readable with internal documentation and meaningful names. In other software development environments, appropriate alternative characteristics should be in evidence.

e) Environment's facilities are used effectively.

Facilities included in the range are:

- editor
- translator
- error tracing.

When carrying out practical exercises to implement features of the selected software environment, the candidate must use its editing and error-tracing facilities effectively and efficiently. The candidate should make effective and efficient use of the translator associated with the environment. For example, the candidate should always make amendments using a text editor rather than re-entering data. They should also be able to use error reporting and any available error-tracing tools to find errors.

This outcome allows candidates to put into practice what they have learned about features of a software development environment and about algorithms in Outcome 2. Therefore, this outcome is likely to be delivered in parallel with Outcome 2 with practical activity being used to consolidate understanding. The practical exercises of this outcome could also be integrated with Outcome 4 at the implementation stage of problem-solving. For example, a problem could be set which requires the implementation of an input validation or the other selected algorithm.

Outcome 4

In this outcome, candidates are required to produce a solution to a problem using a software development environment. In completing this outcome, the candidate will apply the stages of the software development process correctly (PC (a)), in order to produce a working solution to the problem (PC (b)). The solution should be evaluated against given criteria (PC (c)) and a report on the solution will be compiled (PC (d)). Each performance criterion is now described in turn.

a) The stages of the software development process are applied correctly.

The stages contained in the range are:

- analysis
- design
- implementation
- testing
- documentation.

These stages should be applied to a complex problem. Evidence should be as follows:

- analysis - interpret the problem specification by stating the problem in the candidate's own words, identify inputs, outputs and processing operations, as well as making any assumptions
- design - use one design methodology to produce a modular description of the main stages and their sub-refinements, identify the data flow between the modules
- implementation - correct use of syntax and semantics to provide a modular, working solution, effective and efficient use of the editing and error-tracing facilities of the software development environment, and production of a maintainable solution, that is, a solution which contains internal documentation and meaningful variable names
- testing - production of a complete set of test data and systematic testing
- documentation - production of a user guide on how to start up and use the developed software and a technical guide with details of the hardware requirements and software development environment and a structured listing of the solution.

b) A working solution is produced.

In solving the problem, the candidate should provide a working version of the solution to the problem.

c) The solution is evaluated against given criteria.

The criteria contained in the range are:

- user interface
- fitness for purpose
- maintainability
- robustness
- reliability
- portability.

The evaluation of the solution should be comprehensive and include:

- user interface - including reference to visual appeal, screen layout, help screens, instruction screens, prompts for the user and the nature of the HCI and prompts for the user
- fitness for purpose - including consideration of the actual results of applying the test data against the expected results of applying the test data and whether the solution satisfies the problem set
- maintainability - including reference to ease of correction and updating in future
- robustness - consideration of the ability of the developed software to cope with errors during execution without failing
- reliability - consideration of how well the developed software operates without stopping due to design faults
- portability - consideration of the ability of the developed software to be used on a computer other than that for which it was designed.

d) The results of the software development process are communicated clearly and concisely.

The criteria contained in the range are:

- evidence of application of the stages of software development
- evaluation.

It is not necessary that a completely new report be created for evidence of this performance criterion. As far as possible, the report should be compiled from evidence generated for PCs (b) and (c).

In this outcome, the problem to be solved will be non-familiar and complex in nature. It may well involve the implementation of one or more of the algorithms which have been examined under Outcomes 1 and 3. If possible, candidates should have the opportunity to compare solutions which solve the same problem and which are developed in different software development environments.

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

This is the third in a series of units relating to Software Development, and, as such, candidates should be gaining confidence and requiring less direct assistance from teachers and lecturers. The content/context section (above) provides guidance on the delivery of the outcomes. This section focuses on aspects of learning and teaching more generally.

As mentioned previously, although this unit focuses on one selected software development environment, whenever possible, opportunities should be created for candidates to compare features, implementations and solutions in different environments. This should enhance depth of understanding.

The suggestion has been made already that candidates study Outcomes 2 and 3 in parallel. Direct teaching which focuses on the development of a worked example followed by candidates' practising the use of features in similar exercises is a tried and tested delivery model which can contribute to the compilation of a folio of work. Within this model, opportunities can be created also to incorporate 'mini' analysis and design activities.

Learning about the problem-solving stages in parallel with development of the solution to a problem is a possible teaching approach. It is important that candidates are taught and have the opportunity to learn the generic skills required in the software development process. For example, to learn how to analyse a problem, candidates may be encouraged to underline the most important information contained in the problem specification. To learn how to evaluate, candidates must be introduced to the criteria, then they may be given solutions to evaluate or they may be required to compare their products with those of other candidates. Candidates should practise writing documentation before being assessed on the associated skills.

As has been suggested already, there may be benefits in setting a problem-solving activity which requires development and implementation of one or more of the unit's algorithms. This approach could promote relevance and meaningfulness in algorithmic development.

National Unit Specification: support notes (cont)

UNIT Software Development (Higher)

While the distribution of time between the outcomes will vary, candidates may be expected to complete each outcome within the following time scale:

Outcome 1 ..... 10 hours
Outcome 2 ..... 5 hours
Outcome 3 ..... 10 hours
Outcome 4 ..... 15 hours

If this unit is delivered as part of a course (or programme of units) then the course documentation will provide further information on teaching and learning in a course context. This documentation will identify a number of 'themes' to facilitate holistic teaching and learning across the course.

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

Centres may use the instruments of assessment which are considered to be most appropriate. Examples of instruments of assessment which could be used are as follows:

Outcome 1	Questions requiring short and restricted responses which contain explanations of the stages, tools and techniques, characteristics in the range and descriptions of relevant algorithms.
Outcome 2	Questions requiring short and restricted responses on the features of a software development environment, including purposes and means of implementation.
Outcome 3	Compilation of a folio of items selected to provide evidence of implementation of all features in the range and appropriate algorithms. Recorded by means of a checklist.
Outcome 4	An assignment which specifies a suitable problem for solving at this level.

During the work of the unit, candidates should have several opportunities to develop their practical skills and should be assessed at appropriate points. Terminology should be presented in context throughout the unit. Where the candidate is unsuccessful in achieving an outcome, provision should be made for further work and re-assessment.

There are opportunities to integrate assessment of outcomes. For example, Outcomes 2 and 3 could be combined within an assessment item on each feature or small group of features. Also, correct implementation of features and algorithms in the problem solving activity assessment could provide some of the evidence for PC (b) of Outcome 4 and for all performance criteria of Outcome 3. Another possibility is to link evidence for PC (c) of Outcome 1 with evidence for PCs (c) and (d) of Outcome 4.

Written evidence may take various forms including hand-written and word-processed text, or other forms of written communication that are more suited to candidates with physical disabilities. Candidates should not be required to produce a specific form of written text (such as word-processed text) unless this is stipulated within the unit specification or instrument of assessment.

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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).

SCOTTISH QUALIFICATIONS AUTHORITY

National Unit Specification: general information

UNIT Artificial Intelligence (Higher)

NUMBER D101 12

COURSE Computing (Higher)

SUMMARY

This unit is designed to develop an awareness of the scope of artificial intelligence together with knowledge and understanding of some of the concepts and techniques associated with the representation and processing of knowledge. Practical abilities in the use of artificial intelligence software are also developed, as are problem-solving skills relating to knowledge representation and processing.

OUTCOMES

1 Describe the development of artificial intelligence.
2 Describe the techniques used to represent and process knowledge.
3 Solve problems by applying knowledge-processing techniques.
4 Investigate an aspect of artificial intelligence.

RECOMMENDED ENTRY

While entry is at the discretion of the centre, candidates would normally be expected to have attained one or more of the following qualifications (or possess equivalent experience):

- Software Development (Int 2) unit
- Computing course at Intermediate 2 level
- Grade 1 or 2 in Standard Grade Computing Studies.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Unit Specification: general information (cont)

UNIT Artificial Intelligence (Higher)

CREDIT VALUE

1 credit at Higher.

CORE SKILLS

There is no automatic certification of core skills or core skills components in this unit.

Additional information about core skills is published *in Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Unit Specification: statement of standards

UNIT Artificial Intelligence (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

Describe the development of artificial intelligence.

Performance criteria

(a) Significant stages in the development of artificial intelligence are accurately described.
(b) The main areas of development are accurately described.
(c) The philosophical, moral and ethical issues are accurately and concisely explained.
(d) Practical applications of artificial intelligence are accurately and concisely described.

Note on range for the outcome

Stages in development: 1950 to mid 1960s; mid 1960s to mid 1970s; late 1970s onwards.
Issues: difficulties in defining intelligence; modelling aspects of human intelligence; conflicting views on use of artificial intelligence; legal responsibility.

Evidence requirements

Written or oral evidence that the candidate can describe the development of artificial intelligence as detailed in PCs (a) to (d). This may be evidenced by completion of a series of extended-response questions which with reference to PCs (b) and (d) should be related to three areas of development and three practical applications selected from natural language processing, expert systems, computer vision, pattern matching, intelligent robots, heuristics, neural networks, parallel processing.

OUTCOME 2

Describe the techniques used to represent and process knowledge.

Performance criteria

(a) Characteristics of declarative languages are accurately described.
(b) Characteristics of expert systems are accurately described.
(c) Descriptions are concise and technically accurate.

Note on range for the outcome

Characteristics of declarative languages: knowledge base; interrogating the knowledge base; search methods.
Characteristics of expert systems: knowledge base; inference engine; explanatory interface.

Evidence requirements

Written or oral evidence that the candidate can describe the techniques used to represent and process knowledge as detailed in PCs (a) to (c). This may be evidenced by completion of short-response questions relating to the characteristics of declarative languages and the characteristics of expert systems. Answers should include details about a named declarative language and a particular expert system shell.

National Unit Specification: statement of standards (cont)

UNIT Artificial Intelligence (Higher)

OUTCOME 3

Solve problems by applying knowledge-processing techniques.

Performance criteria

(a) Analysis of the problem correctly identifies the nature of the problem and the main features relevant to the solution are clearly described.
(b) Design of solution is clear, complete and in accordance with requirements.
(c) Implementation applies knowledge-processing techniques in accordance with design
(d) Evaluation of solution is complete, concise, and relates to appropriate criteria.

Note on range for the outcome

Main features: boundaries of the problem; assumptions.
Design requirements: simple knowledge base; search method.
Knowledge-processing techniques: declarative language; expert system.
Criteria: functionality; quality; possible enhancements; potential consequences of implementing solution.

Evidence requirements

Written or oral evidence that the candidate can solve simple problems by applying knowledge-processing techniques in two distinct contexts in accordance with PCs (a), (b) and (d).

Performance evidence that the candidate can apply knowledge-processing techniques in two distinct contexts in accordance with PC (c). One implementation should require the use of a declarative language and the other the use of an expert system.

OUTCOME 4

Investigate an aspect of artificial intelligence.

Performance criteria

(a) Strategy for investigation is clear, complete and achievable.
(b) Investigation is carried out systematically in accordance with strategy.
(c) Reporting is clear, concise, accurate and complete.

Note on range for the outcome

There are no specific requirements as to the range of contexts within which this outcome and these performance criteria should be demonstrated. For further guidance on the range of content to be covered, see the support notes.

Evidence requirements

Written or oral evidence that the candidate can identify, investigate and report upon an aspect of artificial intelligence. This may be evidenced by a written report (approximately 800 - 1000 words) as required by PC (c), accompanied by documentary evidence related to PCs (a) and (b).

National Unit Specification: support notes

UNIT Artificial Intelligence (Higher)

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

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

GUIDANCE ON CONTENT AND CONTEXT FOR THIS UNIT

This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Higher level. The course specification provides additional details regarding the delivery and assessment of this unit within a course context. This unit provides elaboration of concepts introduced briefly in Standard Grade Computing Studies and provides progression to units at Advanced Higher level, in particular the optional unit, Artificial Intelligence (AH).

It is anticipated that this unit will be undertaken in a variety of contexts by a wide range of candidates. While previous knowledge and experience of computers is desirable, it is possible for suitably motivated candidates to undertake this unit without previous computing experience.

Corresponding to Outcomes 1–4

This unit is the first of two units relating to artificial intelligence. Reference may be made to specific Internet resources within this unit specification; whenever possible a related URL has been supplied.

Outcome 1

This outcome relates to the development of artificial intelligence and provides the candidate with an overview of the subject area. Candidates should become aware of the difficulties in providing a definitive description of the subject area and, as a consequence, the flexible scope of artificial intelligence.

Candidates are required to describe significant stages in the development of artificial intelligence (PC (a)). The history of artificial intelligence should be outlined, describing the different aims pursued by different generations of artificial intelligence researchers:

- **1950 to mid-1960s:** research focused on creating computers and programs in likeness of humans (for example, early neural networks, perceptron, game playing);
- **mid - 1960s - mid - 1970s:** research focused on knowledge representation and making computers understand (for example ELIZA, GPS, Parry);
- **late 1970s onwards:** research moved towards domain-specific abilities (for example, intelligent robots, natural language processing, expert systems).

Hardware and software developments that either stimulated or were stimulated by each generation of artificial intelligence researchers should be considered along with significant programs.

The main areas of development (including natural language processing, expert systems, computer vision, pattern matching, intelligent robots, heuristics, neural networks and parallel processing) should be described by the candidate (PC (b)). Other areas of development that might be considered include intelligent tutoring and the use of fuzzy logic. (Note that further investigation of one of these main areas of development might provide a suitable choice for the aspect of artificial intelligence investigated for Outcome 4.) In this unit, candidates are expected to have an overall view of each of these areas and are not expected to acquire an in-depth knowledge of any one area. For example, candidates might be expected to explore the area of computer vision to the extent that they are aware that:

- replicating human vision is more complex than was originally thought
- complexities arise as a result of the vast amount of detailed information (colour, distance, texture, prior experience etc) which is processed by humans
- as a purely photographic concept, computer vision can exceed human vision (for example, in measuring distance and position) but cannot extract visual information from a situation and organise and evaluate it with the same reliability as the human eye
- most systems now use digital rather than analogue techniques
- pattern-matching techniques allow a computer vision system to match objects found with a library of standard objects
- uses include enabling industrial robots to make judgements about position, orientation etc.

At this level a detailed knowledge of image acquisition, image processing, image analysis (primal sketch, Waltz' algorithm) and image recognition and understanding would not be expected. Similarly, when considering natural language processing the candidate should become aware of the complexities involved, but should not be expected to consider specific details like parsing.

This outcome offers scope for group discussion of the philosophical, moral and ethical issues associated with intelligent machines (PC (c)). Candidates should be aware of the difficulties of defining intelligence and of contrasting definitions (look up several dictionaries) and that much artificial intelligence research has attempted to model aspects of human intelligence (learning from past experience, creativity, having a conversation, recognising people). Conflicting views on the use of artificial intelligence in a number of areas (for example, medical, legal and financial applications) could be explored by debate, by role play or by interviewing a practitioner. Candidates must also consider who would be held legally responsible if such systems gave incorrect advice and this led, for example, to an incorrect medical diagnosis with fatal consequences, or to financial disaster.

The final part of this outcome concentrates on practical applications of artificial intelligence (PC (d)). Candidates should extend their study of the areas of development listed above and ascertain the extent to which developments have led to practical applications. At this stage, candidates could be guided to explore three of practical applications in greater detail. The areas chosen might reflect the local situation (for example, access to a local company which uses an artificial intelligence-based system to assist with financial advice) or might arise from a search for up-to-date information from the Internet.

Outcome 2

This outcome focuses on techniques for representing and processing knowledge using artificial intelligence techniques. It provides opportunity for practical activity through the use of a declarative programming language and an expert system shell. It is essential that candidates appreciate the difference between these techniques associated with artificial intelligence and conventional computing methods.

Candidates must demonstrate that they can accurately describe characteristics of declarative languages (PC(a)). The representation of simple facts and rules using predicate logic might be a suitable introduction. (At this level, the standard logic symbols should not be used.) This would lead to the use of a declarative programming language (probably either Prolog or Lisp) to represent and process knowledge. Through examining the characteristics of the programming language, candidates should be aware that the knowledge base is represented using facts and rules and that this knowledge base is then interrogated or queried in order to attempt to satisfy the goal. The candidate should explore the programming language to the extent that they have experience of, and can describe negation, recursion and standard list processing functions. Candidates should also explore the concept of inheritance and derive appropriate rules. In addition, the candidate must investigate the search method. Candidates should know the difference between the two types of exhaustive search (depth-first search and breadth-first) and be able to describe the use of backtracking during the search.

Candidates must also be able to describe accurately how an expert system represents and processes knowledge (PC (b)). The roles of the knowledge engineer, the domain expert and the user of an expert system should be distinguished. Candidates should be aware of the functions of the component parts of an expert system: the knowledge base, the explanatory interface and the inference engine. As well as ascertaining how the knowledge base is constructed (facts/rules, frames, database), the candidate should appreciate that to mimic the reasoning of a human expert accurately will require the use of certainty theory where rules have a measure of certainty assigned to them. Study of the inference engine should include forward and backward chaining. With reference to the explanatory interface candidates should consider features of the human computer interface (including menus, graphics, quality of dialogue) and the justification facility (in particular, candidates should be able to distinguish between the two key questions ‘how’ and ‘why’ as applied to justification of reasoning).

The descriptions of the characteristics of declarative languages and expert systems must be concise and technically correct (PC (c)). Thus, the vocabulary specific to declarative languages or expert systems must be used accurately and in the correct context.

Outcome 3

This outcome allows candidates the opportunity to develop problem-solving skills, through the solution of simple problems involving knowledge representation and processing. Candidates should carry out the whole range of problem-solving activities from analysis through design and implementation to evaluation. Some problems should lead to a solution using a declarative programming language; others should be solved using an expert system shell.

Candidates are expected to analyse the problem correctly, identifying the nature of the problem and ensuring that the main features relevant to the solution are described clearly (PC (a)). In their analysis, candidates should ensure that the problem is defined precisely by identifying the problem and stating the boundaries of the problem. Any assumptions should be made explicit.

The design of the solution is expected to be clear and complete and in accordance with requirements (PC (b)). At the design stage, candidates should isolate and represent the essential knowledge. This is likely to involve a simple knowledge base constructed using facts and rules. The appropriate knowledge-processing technique should be selected and the proposed search method noted.

The problem should then be implemented in accordance with the design (PC (c)). The implementation will require the use of a declarative language or an expert system.

Finally, the solution should be evaluated. The evaluation must be complete and concise and appropriate criteria must be considered (PC (d)). The evaluation must consider the following:

- functionality - Does the solution do what it is meant to? Does it satisfy the original problem? Could the answer have been found more readily by some other means?
- quality of the solution - Is the solution efficient? Is it valid?
- possible enhancements - How could the solution be improved? Could the problem and/or the solution be generalised?
- possible consequences of implementing the solution - Are there moral or ethical implications associated with applying artificial intelligence techniques to this problem? Are there legal considerations?

Outcome 4

This outcome provides candidates with an opportunity to investigate an aspect of artificial intelligence. Ideally, candidates will have been stimulated by the work undertaken for Outcomes 1, 2 and 3 and each candidate will be ready to select an aspect of artificial intelligence in which he/she is interested. The aspect selected might involve:

- exploring a earlier topic in greater depth (for example, the implications of using expert systems in medicine)
- extending earlier work to investigate a related aspect (for example, extend work on natural language processing to include parsing)
- investigating an aspect which links artificial intelligence to a topic covered in other computing units completed by a candidate (for example, an investigation into the hardware and software limitations which have restricted developments in computer vision).

National Unit Specification: support notes (cont)

UNIT Artificial Intelligence (Higher)

Note that the investigation for this unit must focus on artificial intelligence and not on other computing topics. Thus, an investigation which simply compared a declarative and procedural language in general terms would be more appropriate as a topic for investigation in the optional Computer Programming (H) unit rather than in this unit.

At the analysis and design stage, candidates are required to devise a strategy for the investigation and ensure that it is clear, complete and achievable (PC (a)). The candidate should:

- interpret the initial problem definition in order to establish the nature of the investigation
- negotiate with the teacher/lecturer an agreed definition of the task and the criteria that would identify a successful outcome
- identify the main stages to be carried out and the strategy to be adopted at each stage
- recognise the range and nature of any new information and techniques to be acquired
- identify suitable resources.

Candidates should be reminded that the topic chosen should be sufficiently focused so that the investigation, including the final report, can be completed in approximately 10 hours. At this stage candidates should ensure that the initial plans and more detailed proposals are documented adequately so that evidence related to this performance criterion is available along with the final report.

At the implementation stage the investigation must be carried out systematically in accordance with the strategy (PC (b)). The candidate should be able to:

- follow through the planned stages systematically
- extract and collate the relevant information
- apply knowledge and techniques successfully
- draw conclusions which take account of all the important factors relating to the task.

Again, candidates should ensure that details about how the investigation is carried out are documented adequately so that evidence related to this performance criterion is available along with the final report. This might include a list of reference books consulted, web sites visited, questions prepared prior to discussion with a university researcher or an expert system user.

The final report is required to be clear, concise, accurate and complete (PC (c)).

The written reports produced by the candidate should be technically accurate, clearly stated and complete. Candidates should be discouraged from spending an undue proportion of the time allocated on presentation, as it is the content of the report which will be assessed not its superficial appearance.

National Unit Specification: support notes (cont)

UNIT Artificial Intelligence (Higher)

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

A candidate-centred, resource-based approach is recommended. While the distribution of time between the outcomes will vary, candidates might be expected to complete each outcome within the following time scale:

Outcome 1 ..... 6 hours
Outcome 2 ..... 12 hours
Outcome 3 ..... 12 hours
Outcome 4 ..... 10 hours

A range of software will be required to deliver this unit. Where possible, candidates should have access to software which illustrates some of the developments associated with artificial intelligence which are considered for Outcome 1. For example, there are many popular versions of the renowned ELIZA 'conversation'. For Outcome 2, candidates should have access to a declarative language (for example, Prolog or Lisp) and an expert system shell. Similarly, access to a declarative language and an expert system shell would enhance much of the work for Outcome 3, by making the link between knowledge-processing techniques and possible computer-based solutions to problems explicit.

While much of Outcome 1 could be treated as theory, candidates are likely to be stimulated by consideration of the portrayal of intelligent computers in popular media and through the use of alternative texts such as Isaac Asimov's robot stories.

For all outcomes, learning and teaching will be enhanced if as well as having access to standard artificial intelligence textbooks candidates can readily refer to current computing periodicals. Many of these regularly feature articles on recent developments related to artificial intelligence. Similarly TV programmes often include similar reports. Interesting and up-to-date material will also be found on the Internet.

A visit to a local company which uses an expert system (for example, giving financial advice, or a help desk for assistance with fault diagnosis) or a talk about current developments from a researcher (these may be based in universities/colleges or in commercial/industrial organisations) would help candidates become aware of the difficulties encountered in the field of artificial intelligence, the progress made in recent years and the stimuli for further developments.

There is scope for integrative work across the outcomes and it is important that candidates do not compartmentalise the work for each outcome. For example, expert systems is one of the areas of development in Outcome 1. The work in Outcomes 2 and 3 builds on candidates' understanding of expert systems as an area of development from Outcome 1.

Outcome 4 is likely to be done after the other outcomes. However, it is essential that from an early stage candidates are aware that an investigation is to be undertaken. If they are alerted to this from the start it should be much easier for them to focus on an aspect of artificial intelligence which would be a suitable topic for the investigation.

National Unit Specification: support notes (cont)

UNIT Artificial Intelligence (Higher)

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

Centres may use the instruments of assessment which are considered to be the most appropriate. Examples of instruments of assessment (IA) which could be used are as follows:

Outcome 1	IAI: Written assignment (extended-response questions), requiring descriptions about each of the three significant stages in the development of artificial intelligence research. 1A2: Written assignment (extended-response questions), describing three areas of artificial intelligence research and the resulting practical applications. IA3: Written assignment (extended-response questions), explaining philosophical, moral and ethical issues. Alternatively, this assignment could be presented orally as a debate with two candidates outlining the contrasting views and challenging the other view. (A video or audio recording could be made as a record of this.)
Outcome 2	IA I: Written assignment (short-response questions), describing the characteristics of declarative languages and including specific details about a named language. 1A2: Written assignment (short-response questions), describing the characteristics of expert systems and including specific details about a particular expert system shell.
Outcome 3	1A1 and 1A2: Two practical assignments set in two distinct contexts, where, for each, the candidate analyses the problem, designs the solution, implements and then evaluates it. One assignment should require the use of a declarative language and the other an expert system shell.
Outcome 4	IA I: A written report of approximately 800 - 1000 words which details the results of the investigation. The final report should be accompanied by documented evidence outlining the original analysis and design and listing the sources consulted at the implementation stage.

During the work of the unit, candidates should have repeated opportunities to extend their knowledge and understanding of artificial intelligence, to apply their problem-solving skills to topics related to artificial intelligence and to develop associated practical skills. Terminology should be presented in context throughout the unit. Where the candidate is unsuccessful in achieving an outcome, provision should be made for remediation and re-assessment.

There are opportunities to integrate assessment within this unit. For example, the aspect of artificial intelligence chosen for the investigation in Outcome 4 might be an extension of the content of Outcome 1. In this case, a more extended report than that suggested for Outcome 4 could be submitted which would also satisfy the evidence requirements for Outcome 1. Similarly, it would be possible to combine the assessment of Outcomes 2 and 3 by requiring the solution to problems being solved for Outcome 3 to include descriptions about the characteristics of the selected technique for representing and processing knowledge. For example, when documenting a solution to a fault diagnosis system, the candidate should accurately describe characteristics of an expert system.

Written evidence may take various forms including hand-written and word-processed text, or other forms of written communication that are more suited to candidates with physical disabilities. Candidates should not be required to produce a specific form of written text (such as word-processed text) unless this is stipulated within the unit specification or instrument of assessment.

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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).

SCOTTISH QUALIFICATIONS AUTHORITY

National Unit Specification: general information

UNIT Computer Programming (Higher)

NUMBER D100 12

COURSE Computing (Higher)

SUMMARY

This unit introduces candidates to the algorithms and data structures used in the processing of more complex, structured data and develops the ability to use a systematic program design methodology to produce complex programs. Candidates also develop skills in the testing and debugging of computer programs.

OUTCOMES

1 Explain aspects of the software development process.
2 Explain features of a software development environment.
3 Implement features and use facilities of a software development environment.
4 Produce a solution to a problem using a software development environment.
5 Investigate an aspect of computer programming.

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 experience):

- Software Development (H) unit
- Computing course at Intermediate 2 level
- Grade 1 or 2 at Standard Grade in Computing Studies.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Unit Specification: general information (cont)

UNIT Computer Programming (Higher)

CREDIT VALUE

1 credit at Higher.

CORE SKILLS

There is no automatic certification of core skills or core skills components in this unit.

Additional information about core skills is published *in Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Unit Specification: statement of standards

UNIT Computer Programming (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

Explain aspects of the software development process.

Performance criteria

(a) The debugging stage of the software development process is explained accurately and concisely.
(b) Tools and techniques of the software development process are explained accurately.
(c) Efficient use of resources by software is explained accurately.
(d) Algorithms are designed and described accurately.

Note on range for the outcome

Tools and techniques: editors; debuggers.
Resources: memory; processor time.
Algorithms: binary search; sort; list operations; counting occurrences.

Evidence requirements

Written or oral evidence that the candidate can explain aspects of the software development process as detailed in PCs (a) to (b) for all classes in the range and how software can be efficient in its use of memory and processor time (PC (c). Evidence should cover dry run, trace tables, break points, and at least one from insert, delete and append item in lists. Written or diagrammatic evidence that the candidate has designed and described the algorithms detailed in PC (d) for all classes in the range.

OUTCOME 2

Explain features of a software development environment.

Performance criteria

(a) Features of a software development environment are defined correctly.
(b) Purposes of features are explained correctly.
(c) Means of implementing features are explained correctly.

Note on range for the outcome

Features: data structures; file handling.

Evidence requirements

Written or oral evidence that the candidate can explain features of a software development environment as detailed in PCs (a) to (c) for all classes in the range. Evidence should cover at least one data structures feature from: two-dimensional arrays, records, queues and stacks; and at least one file handling feature from: file creation, file operation and sequential file access.

National Unit Specification: statement of standards (cont)

UNIT Computer Programming (Higher)

OUTCOME 3

Implement features and use facilities of a software development environment.

Performance criteria

(a) Syntax and semantics are used correctly.
(b) Implementation of the features of the software development environment is complete.
(c) Algorithms are implemented correctly.
(d) Characteristics which allow for maintenance of implementations are used appropriately.
(e) The environment's facilities are used efficiently and effectively.

Note on range for the outcome

Features: data structures; file handling.
Algorithms: binary search; sort; list operations; counting occurrences.
Characteristics: modularity; internal documentation; meaningful names.
Facilities: editors; translators; debuggers.

Evidence requirements

Performance evidence that the candidate can implement features and use facilities of a software development environment as detailed in PCs (a) to (e) for all classes in the range. Evidence should cover use of one-dimensional arrays in parallel, two-dimensional arrays and file handling. Also included would be one list operation from the operations: insert, delete, append.

OUTCOME 4

Produce a solution to a problem using a software development environment.

Performance criteria

(a) The stages of the software development process are applied correctly.
(b) Implementation of the solution to the problem is complete.
(c) The solution is evaluated against given criteria.
(d) The results of the software development process are communicated clearly and concisely.

Note on range for the outcome

Stages: analysis; design; debugging; implementation; testing; documentation.
Criteria: user interface; fitness for purpose; maintainability; robustness; reliability; portability; efficiency of use of memory and of processor time.

Evidence requirements

Performance evidence that the candidate can produce a solution to a problem using, a software development environment as detailed in PCs (a) and (b) for the relevant class in the range. The solution should include both user and technical documentation.

Written or oral evidence that the candidate can evaluate and report on a solution as detailed in PCs (c) and (d) for relevant classes in the range.

National Unit Specification: statement of standards

UNIT Computer Programming (Higher)

OUTCOME 5

Investigate an aspect of computer programming.

Performance criteria

(a) Strategy for investigation is clear, complete and achievable
(b) Investigation is carried out systematically in accordance with strategy.
(c) Reporting is clear, concise, accurate and complete.

Note on range for the outcome

There are no specific requirements as to the range of contexts within which the outcomes and performance criteria should be demonstrated. For further guidance on the range of content to be covered, see the support notes.

Evidence requirements

Evidence that the candidate can identify, investigate and report upon an aspect of computer programming. This may be evidenced by a written report (approximately 800 - 1000 words).

National Unit Specification: support notes

UNIT Computer Programming (Higher)

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

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

GUIDANCE ON CONTENT AND CONTEXT FOR THIS UNIT

This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Higher level. The details of content given earlier provide information on the delivery and assessment of this unit within a course context.

Corresponding to Outcomes 1–5

This unit is the fourth in a series of units related to Software Development and other units are available at Intermediate 1, Intermediate 2, Higher and Advanced Higher levels. It should be noted that Computer Programming (H) is an optional unit which builds on learning in the core of Software Development (H). As such, this unit is intended to consolidate and extend earlier learning in this area.

Progression through this series of Software Development units is shown in the relevant progression matrix (see Subject Guide). It should be noted that four main themes provide the structure for progression. Of these themes, two are associated with knowledge and understanding of computing facts and concepts and two with practical skills and abilities. The former themes are: software development process (stages, techniques and tools, characteristics, algorithms); and software development environment features. The latter themes are: working in a software development environment (Implementation of features, implementation of algorithms, implementation of maintenance characteristics, use of environment's facilities); and problem-solving (analysis, design, implementation, testing, documentation, evaluation). It should be noted that Computer Programming (H) includes an investigation, through which candidates are expected to extend their knowledge and skills base.

Progression through the Software Development units is promoted in a number of ways. Introduction of new aspects increases breadth of study through units. Where the same aspects are included in units, depth of study is the key to progression. This depth can be manifested in different ways, for example: the level of cognitive skill becomes higher (eg recall, describe, explain); the nature of response is increasingly demanding (eg short to extended response); the level of difficulty of questions/problems (eg simple/complex, familiar/unfamiliar contexts) increases. In all units, candidates are expected to use correctly and in context terminology appropriate to the level of study. For example, within the context of software development, the term 'user interface' refers to factors that relate to the interaction between the user and the developed software solution. Where appropriate, guidance is given about the appropriate level of treatment of terms within the unit.

When studying a core course unit at a particular level, it is assumed that candidates are able to use knowledge and skills which are included in mandatory units (but not optional units) in the series at lower levels. Study of an optional unit assumes knowledge and skills in the mandatory unit at the same level. In the case of Computer Programming (H), prior knowledge and skills are assumed from Software Development (H).

The term *'software development environment'* refers to any environment whose features and facilities can be used to produce solutions to problems. An integrated environment with a 'programming' language (using the term in its very broadest sense) and facilities such as text editor and debuggers would be ideal. An imperative language, a declarative language, an applications package with a high functional capability, an object-oriented language, a scripting language, or an authoring environment are all suitable. The features contained in the statement of standards are intended to be sufficiently open to facilitate delivery of the Software Development units in this wide range of environments. The support notes are exemplified in terms of a procedural high-level language. Implementations in other environments should be broadly compatible.

At Higher level, in-depth knowledge of data structures and algorithms along with associated skills should be developed within the context of one selected software development environment. Given that the rationale for this unit is to provide the opportunity for enthusiasts to concentrate on programming then it is acknowledged that some environments may be better suited to its delivery than others.

Outcome 1

This outcome relates to knowledge and understanding of aspects of the software development process. In this unit, the candidate is expected to explain the debugging stage of the software development process accurately and concisely (PC (a)) and explain the tools and techniques of the software development process accurately (PC (b)). In addition, efficient use of resources must be explained (PC (c)) and algorithms be designed and described accurately (PC (d)). Further details about each performance criterion follow.

a) The debugging stage of the software development process is explained accurately and concisely.

Candidates are expected to explain debugging in terms of detection, location and correction of faults (or bugs) in a program. Explanations should include reference to the place of debugging at different stages in the software development process.

b) Tools and techniques of the software development process are explained accurately.

The tools and techniques detailed in the range are:

- editors
- debugging.

Candidates are expected to explain accurately:

- editors - programs which allow the user to input, inspect and alter program files. Most editors have special features which make the programmer's task easier
- debugging - the detection, location and correction of faults (or bugs) in a program using debugging tools such as: dry run and trace table - a manual work through of a section of a program to identify the instruction executed and the contents of each variable; and breakpoint - a position within a program where the program is halted as an aid to debugging.

c) Efficient use of resources by software is explained accurately.

The resources relate to:

- memory
- processor time.

The candidate is expected to be able to explain:

- efficiency of memory usage - the memory utilised by the program should not require memory resources disproportionate to the scale of the program
- efficiency of processor time - algorithms should not utilise storage or processing time disproportionate to their function.

d) Algorithms are designed and described accurately.

The algorithms contained in the range are:

- counting occurrences
- binary search
- sort
- list operations.

The candidate should be able to design and describe each of the listed algorithms. List operations include inserting, deleting and appending items in lists. It is emphasised that there is no requirement to implement all of these.

In order to enhance the acquisition of knowledge and understanding, it would be highly desirable for candidates to experience the use of debugging and editing, tools. In the case of debugging tools, small-scale practical examples might be set up for the candidate to debug. A range of features, such as find and search and replace, which are used in program editors could also be examined using practical examples.

The comparison of internal sorting algorithms would illustrate efficiency of memory usage and processing time. The use of real variable data types against integer variable data types would demonstrate one aspect of efficiency of memory usage as would the declaration of string over long string variables and unnecessarily large arrays. Illustrative examples of algorithms in action would provide a useful basis on which to develop knowledge and understanding.

Outcome 2

This outcome relates to knowledge and understanding of features of a software development environment. Candidates are expected to define correctly features of a software development environment (PC (a)), explain correctly the purposes of these features (PC (b)) and explain correctly how the features are implemented (PC (c)) in a selected environment. The emphasis is on data structures. Further details about each performance criterion follow.

a) Features of a software development environment are defined correctly.

The features listed in the range are:

- data structures
- file handling.

At this level, the candidate must be able to give comprehensive definitions of features. The candidate is expected to be able to define the following data structures:

- two-dimensional arrays
- records
- queues
- stacks.

Note that the candidate will not have to implement all these structures. The candidate should be able to define file-handling with reference to:

- file creation
- file operations
- sequential means of access.

b) Purposes of features are explained correctly.

The features listed in PC (a) apply also in this context. The candidate is expected to explain in some detail the purposes of features, with illustrative examples of when their use is appropriate.

c) Means of implementing features are explained correctly.

The features listed in PC (a) apply also in this context. The candidate is expected to explain the means of implementing features in appropriate software development environments.

This outcome provides candidates with the opportunity to gain insight into data structure features of a software development environment. Content is likely to be delivered alongside that of Outcome 3 with knowledge and understanding being reinforced by practical activity. Candidates could be introduced to new features one at a time or in small related groups, then carry out simple practical exercises, then answer questions related to PCs (a) to (c). Practical exercises might involve modification of given implementations or implementation of relevant features from scratch. It would be beneficial to learning if candidates could compare implementations of features in different software development environments.

Outcome 3

This outcome relates to practical skills and abilities associated with use of fundamental features and facilities of a software development environment and the implementation of algorithms. Candidates are expected to use syntax and semantics correctly (PC (a)) when producing working implementations of particular features (PC (b)) and selected algorithms (PC (c)). All implementations should be maintainable (PC (d)). When carrying out practical exercises, candidates should use effectively and efficiently editing, error-tracing and debugging facilities of the software development environment (PC (c)). Further details about each performance criterion follow.

a) Syntax and semantics are used correctly.

Features included in the range are:

- data structures
- file handling.

At this level, the candidate should be able to use correct syntax and semantics for the range features in one software development environment, particularly syntax associated with:

- one-dimensional arrays used in parallel
- two-dimensional arrays
- file creation
- file operations
- sequential file access.

b) Working implementations of features are produced.

The features listed in PC (a) apply also in this context. As well as using correct syntax, candidates are expected to be able to complete practical exercises which demonstrate that they can use correctly the range features in one software development environment. For example, in a procedural language, the candidate is expected to use correctly features detailed above. Whenever possible, practical exercises should cover several features or aspects of the features.

c) Algorithms are implemented correctly.

The algorithms contained in the range are:

- counting occurrences
- binary search
- sort
- list operations.

In the selected software development environment, candidates must implement each of the algorithms in the range, including at least one list operation from:

- insert
- delete
- append.

The implementation may be based on an algorithm designed and described by the candidate or on a design provided by the teacher or lecturer.

d) Characteristics which allow for maintenance of implementations are used appropriately.

Characteristics:

- modularity
- internal documentation
- meaningful variable names.

Candidates are expected to be able to implement features in a way which facilitates correction and updating in future. For example, in the context of a procedural language, candidates should produce programs which have a modular structure and which are readable with internal documentation and meaningful names. In other software development environments, appropriate alternative characteristics should be in evidence.

e) The environment's facilities are used effectively and efficiently.

Facilities included in the range are:

- editing
- translation
- debugging.

When carrying out practical exercises to implement features of the selected software environment, candidates must use its editing, translation and debugging facilities effectively and efficiently. For example, candidates should always make amendments using a text editor rather than re-entering data. They should also be able to use error reporting and any available error-tracing tools to find errors. At appropriate stages, they should demonstrate ability to use debugging tools and techniques such as dry run, trace tables and breakpoints.

National Unit Specification: support notes (cont)

UNIT Computer Programming (Higher)

This outcome allows candidates to put into practice what they have learned about features of a software development environment and about algorithms in Outcome 2. Therefore, this outcome is likely to be delivered in parallel with Outcome 2 with practical activity being used to consolidate understanding. The practical exercises of this outcome could also be integrated with Outcome 4 at the implementation stage of problem-solving. For example, a problem could be set which requires the implementation of file handling and a search algorithm.

Outcome 4

In this outcome, candidates are required to produce a solution to a problem using a software development environment. In completing this outcome, the candidate will apply the stages of the software development process correctly (PC (a)), in order to produce a working solution to the problem (PC (b)). The solution should be evaluated against given criteria (PC (c)) and a report on the solution will be compiled (PC (d)). Further details about each performance criterion follow.

a) The stages of the software development process are applied correctly.

The stages contained in the range are:

- analysis
- design
- debugging
- implementation
- testing
- documentation.

These stages should be applied to a complex problem. Evidence should be as follows:

- analysis - interpret the problem specification by stating the problem in the candidate's own words, identify inputs, outputs and processing operations, as well as making any assumptions
- design - use one design methodology to produce a modular description of the main stages and their sub-refinements, identify the data flow between the modules
- debugging - using appropriate tools and techniques, locate and correct errors in the design of algorithms
- implementation - correct use of syntax to provide a modular, working solution, effective and efficient use of the editing and error-tracing facilities of the software development environment, and production of a maintainable solution, that is, a solution which contains internal documentation and meaningful variable names
- testing - production of a complete set of test data and systematic testing
- documentation - production of a user guide on how to start up and use the developed software, and a technical guide with details of the hardware requirements and software development environment, and a structured listing of the solution.

b) A working solution is produced.

In solving the problem, the candidate should provide a working version of tile solution to the problem.

c) The solution is evaluated against given criteria.

The criteria contained in the range are:

- user interface
- fitness for purpose
- maintainability
- robustness
- reliability
- portability
- efficiency of memory use
- efficiency of processor time.

The evaluation of the solution should be comprehensive and include:

- user interface - including reference to visual appeal, screen layout, help screens, instruction screens, prompts for the user and the nature of the HCI
- fitness for purpose - including consideration of the actual results of applying the test data against the expected results of applying the test data and whether the solution satisfies the problem set
- maintainability - including reference to case of correction and updating in future
- robustness - consideration of the ability of the developed software to cope with errors during execution without failing
- reliability - consideration of how well the developed software operates without stopping due to design faults
- portability - consideration of the ability of the developed software to be used on a computer other than that for which it was designed
- efficiency of memory use - consideration of the effect of the data structures employed in the solution on the available memory
- efficiency of processor time - consideration of whether algorithms have been designed and implemented in a manner which does not compromise the speed of execution of the solution or the efficiency of memory use.

d) The results of the software development process are communicated clearly and concisely.

It is not necessary that a completely new report be created for evidence of this performance criterion. As far as possible, the report should be compiled from evidence generated for PCs (b) and (c).

In this outcome, the problem to be solved will be non-familiar and complex in nature. It may well involve the implementation of one or more of the features and algorithms that have been examined under Outcomes 1, 2 and 3.

National Unit Specification: support notes (cont)

UNIT Computer Programming (Higher)

Outcome 5

This outcome relates to the development and use of investigative skills. Candidates are expected to devise a suitable investigation strategy (PC (a)), apply the strategy systematically when carrying out the investigation (PC (b)) and produce a suitable report (PC (c)). Further details about each performance criterion follow.

a) Strategy for investigation is clear, complete and achievable.

The candidate should be able to:

- interpret the initial problem definition in order to establish the nature of the investigation
- negotiate with the teacher or lecturer an agreed definition of the task and the criteria which define a successful outcome
- identify the main stages to be carried out and the strategy to be adopted at each stage.

b) The investigation is carried out systematically in accordance with strategy.

The candidate should be able to:

- follow through the planned stages systematically
- extract and collate the relevant information
- apply knowledge and techniques successfully
- draw conclusions which take account of all the important factors relating to the task.

c) Reporting is clear, concise, accurate and complete.

The candidate should be able to produce a report of 800 to 1000 words which is technically accurate, clearly stated and complete.

This outcome provides candidates with the opportunity to investigate an aspect of computer programming that is new and relevant to study in this unit. For example, candidates might be given a task that requires that they investigate file-handling or searching techniques. The investigation might involve the use of manuals, on-line help, exemplars and practical experiment. The investigative activity might be incorporated in the problem-solving activity of Outcome 4. The investigation report is likely to contain evidence for Outcome 2.

National Unit Specification: support notes (cont)

UNIT Computer Programming (Higher)

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

This is the fourth in a series of units relating to Software Development, an optional unit designed for programming enthusiasts and, therefore, candidates should be confident and requiring less direct assistance from teachers and lecturers. The content/context section (above) provides guidance on the delivery of the outcomes. This section focuses on aspects of learning and teaching more generally.

The suggestion has been made already that candidates study Outcomes 2 and 3 in parallel. Direct teaching which focuses on the development of a worked example followed by candidates' practising the use of features in similar exercises is a tried and tested delivery model which can contribute to the compilation of a folio of work. Within this model, opportunities can be created also to incorporate 'mini' analysis and design activities.

Learning about the problem-solving stages in parallel with development of the solution to a problem is a possible teaching approach. It is important that candidates are taught and have the opportunity to learn the generic skills required in the software development process. For example, to learn how to analyse a problem, candidates may be encouraged to underline the most important information contained in the problem specification. To learn how to evaluate, candidates must be introduced to the criteria, then they may be given solutions to evaluate or they may be required to compare their products with those of other candidates. Candidates should practise writing documentation before being assessed on the associated skills.

As has been suggested already, there may be benefits in setting a problem-solving activity which requires development and implementation of one or more of the unit's algorithms. This approach could promote relevance and meaningfulness in algorithmic development. In addition, in this unit the subject for investigation is likely to be an algorithm or some of the relevant features of the software development environment.

While the distribution of time between the outcomes will vary, candidates may be expected to complete each outcome within the following time scale:

Outcome 1 ..... 8 hours
Outcome 2 ..... 4 hours
Outcome 3 ..... 8 hours
Outcome 4 ..... 10 hours
Outcome 5 ..... 10 hours

If this unit is delivered as part of a course (or programme of units), then the course documentation will provide further information on teaching and learning in a course context. This documentation will identify a number of 'themes' to facilitate holistic teaching and learning across the course.

National Unit Specification: support notes (cont)

UNIT Computer Programming (Higher)

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

Centres may use the instruments of assessment which are considered to be most appropriate. Examples of instruments of assessment which could be used are as follows:

Outcome 1	Questions requiring short and restricted responses which contain explanations of the stages, tools and techniques, characteristics in the range and descriptions of relevant algorithms.
Outcome 2	Questions requiring short and restricted responses on the features of a software development environment, including purposes and means of implementation.
Outcome 3	Compilation of a folio of items selected to provide evidence of implementation of all features in the range and appropriate algorithms. Recorded by means of a checklist.
Outcome 4	An assignment which specifies a suitable problem for solving at this level.
Outcome 5	An assignment to investigate an aspect of computer programming.

During the work of the unit, candidates should have several opportunities to develop their practical skills and should be assessed at appropriate points. Terminology should be presented in context throughout the module. Where the candidate is unsuccessful in achieving an outcome, provision should be made for further work and reassessment.

There are opportunities to integrate assessment of outcomes. For example, Outcomes 2 and 3 could be combined within an assessment item on each feature or small group of features. Also, correct implementation of features and algorithms in the problem-solving activity assessment could provide some of the evidence for PC (b) of Outcome 4 and for all performance criteria of Outcome 3. Another possibility is to link evidence for PC (c) of Outcome 1 with evidence for PCs (c) and (d) of Outcome 4. Certainly, it is desirable to use for other outcomes evidence generated through the investigation of Outcome 5. For example, the investigation report could contain evidence for Outcome 2 and investigation activity could have produced evidence for parts of Outcome 3.

Written evidence may take various forms including hand-written and word-processed text, or other forms of written communication that are more suited to candidates with physical disabilities. Candidates should not be required to produce a specific form of written text (such as word-processed text) unless this is stipulated within the unit specification or instrument of assessment.

National Unit Specification: support notes (cont)

UNIT Computer Programming (Higher)

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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).

SCOTTISH QUALIFICATIONS AUTHORITY

National Unit Specification: general information

UNIT Computer Networking (Higher)

NUMBER D099 12

COURSE Computing (Higher)

SUMMARY

This unit is designed to develop knowledge and understanding of computer networks, including internet-working and to provide an opportunity to apply this knowledge through the use of contemporary networked systems.

OUTCOMES

1 Describe the applications and benefits of computer networks.
2 Describe the characteristics of computer networks and internetworks.
3 Use the facilities provided by a computer network.
4 Investigate an aspect of internetworking

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 experience):

- Computer Systems (H) unit
- Computing course at Intermediate 2 level
- Grade 1 or 2 at Standard Grade in Computing Studies.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Unit Specification: general information (cont)

UNIT Computer Networking (Higher)

CREDIT VALUE

1 credit at Higher.

CORE SKILLS

There is no automatic certification of core skills or core skills components in this unit.

Additional information about core skills is published *in Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Unit Specification: statement of standards

UNIT Computer Networking (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

Describe the applications and benefits of computer networks.

Performance criteria

(a) The economic and technical factors that have led to the development of computer networks are correctly described.
(b) The main benefits of computer networks are correctly described.
(c) The main applications of computer networks are correctly described.
(d) The social, ethical and legal implications of computer networks are accurately explained.

Note on range for the outcome

Computer networks: local-area networks (including intranets); wide-area networks (including the Internet).

Evidence requirements

Written or oral evidence that the candidate can describe the applications and benefits of computer networks as detailed in the performance criteria for all classes in the range. While the benefits of computer networks may be described in generic terms, at least two specific applications of local-area networks (including intranets) and two applications of wide-area networks (including the Internet) must be described.

OUTCOME 2

Describe the characteristics of computer networks and internet-working.

Performance criteria

(a) The major characteristics of computer networks are correctly described.
(b) The major characteristics of internetworks are correctly described.
(c) Contemporary examples of each type of network are correctly described.
(d) The importance of standards is accurately explained and significant national and international standards relating to internetworking are correctly described.

Note on range for the outcome

Computer networks: local-area network (including intranets); wide-area network (including the Internet); peer-to-peer; client-server.
Characteristics: hardware (including topology); software; communication (including, protocols and bandwidth); functionality; security.

Evidence requirements

Written or oral evidence that the candidate can describe the characteristics of computer networks and internetworks as detailed in PCs (a), (b) and (d) for all classes in the range and give at least two contemporary examples of a local area network (including intranets) and at least two contemporary examples of a wide area network (including the Internet).

National Unit Specification: statement of standards (cont)

UNIT Computer Networking (Higher)

OUTCOME 3

Use the facilities provided by a computer network.

Performance criteria

(a) The characteristics of a specific network are known.
(b) The facilities provided by a specific network are known.
(c) The network is used efficiently and effectively.
(d) Network etiquette is known and observed.
(e) Security and legal requirements are observed.

Note on range for the outcome

Characteristics: hardware (including topology); software; communication (including protocols and bandwidth); functional; security.

Evidence requirements

Written or oral evidence that the candidate can describe the characteristics and facilities provided by a specific computer network as detailed in PCs (a) and (b).

Performance evidence that the candidate can use the facilities provided by a computer network as detailed in PCs (c) and (e).

Written or oral evidence that the candidate knows network etiquette as detailed in PC (d).

OUTCOME 4

Investigate an aspect of computer networking.

Performance criteria

(a) Strategy for the investigation is clear, complete and achievable
(b) The investigation is carried out systematically in accordance with strategy.
(c) Reporting is clear, concise, accurate and complete.

Note on range for the outcome

There are no specific requirements as to the range of contexts within which the outcomes and performance criteria should be demonstrated. For further guidance on the range of content to be covered, see the support notes.

Evidence requirements

Written or oral evidence that the candidate can investigate an aspect of computer networking, as detailed in the performance criteria. This will be in the form of a written report detailing the finding of the investigation into an aspect of computer networking.

National Unit Specification: support notes

UNIT Computer Networking (Higher)

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

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

GUIDANCE ON CONTENT AND CONTEXT FOR THIS UNIT

This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Higher level. The details of content given earlier provide information on the delivery and assessment of this unit within a course context. The following details provide information on the delivery and assessment of this unit in a stand-alone context.

Corresponding to Outcomes 1–4

This unit is a mix of cognitive and practical outcomes and seeks to reinforce learning by applying knowledge and understanding in the context of a specific network system.

Outcome 1

This outcome relates to the applications and benefits of computer networks. The economic and technical factors that have led to the growth in computer networks must be described (PC (a)). Economic factors include the decline in the cost of communication and telecommunication technologies and services, shared access to expensive equipment (such as mass storage or printing facilities), the geographic spread of contemporary organisations, increased competition within the public and private sectors and increased demand for up-to-date information. Technical factors include advances in computer technology, new data transmission media and methods, improved quality of networking software, the establishment of international standards for communications and networking (such as the OSI reference model) and the adoption of Internet technology (such as TCP/IP). The last factor should be emphasised as the reason for the recent dramatic increase in connectivity.

The benefits of computer networks (PC (b)) include shared resources (such as printer sharing across a LAN), improved communications (such as the use of e-mail across the Internet) and access to corporate or global information (using an internal or external web) and file transfer capability.

The applications of networks (PC (c)) can be considered under two headings - private sector applications (such as banking and travel) and public sector applications (such as education and health). While the benefits of computer networks may be described in generic terms, the applications must be related to specific types of network system. The selected applications must include the applications of intranets (such as internal information systems) and the applications of the Internet (such as access to the World Wide Web). Other applications of local area networks (LANs) include the use of peer-to-peer systems to share peripherals, the use of client-server systems to provide company-wide e-mail services, and wide-area networks (WANs) to allow file transfer between remote sites and to provide video conference facilities.

The social, ethical and legal implications of computer networks must also be considered (PC (d)). The social implications relate to the effects of networking on the individual and society and encompass issues relating to personal privacy, censorship, disconnection and social isolation. There are currently fears that the spread of computer networks will compromise individual privacy by enabling government agencies and private organisations to monitor individuals. At the time of writing, the issue of censorship on the Internet is hotly debated. The information revolution may result in the 'disconnection' of people who do not have access to the information superhighway resulting in 'information-rich' and 'information-poor' individuals. Social isolation could result from individuals or families withdrawing from society through their use of communication technology within the home. The ethical implications of computer networks relate to the ways in which networks can affect standards of conduct within a wide range of professions such as education, health and finance. Legal implications relate to the ways in which increasing connectivity and globalisation affects national laws. For example, at the time of writing the status of national laws relating to pornography are uncertain when such material is hosted on servers outside national boundaries.

Outcome 2

This outcome relates to the characteristics of networks and internetworks. The main characteristics are hardware, software, communication, functionality and security. Hardware characteristics include topology, servers, workstations, network interface, cabling and concentrator. Candidates should be familiar with the star, ring and bus topologies and file, print and communication servers. Software characteristics include the network operating system, the workstation operating system and the network shell requester or redirector. Communication characteristics include configurations, protocols and bandwidths. Candidates should be familiar with peer-to-peer and client-server configurations. Functional characteristics relate to the services and resources provided by the network. Appropriate services include resource sharing, file transfer, e-mail and search services; network resources are the information and files made available over the network. Security characteristics include user log-in, account control and service/resource access control. The level of treatment of any one characteristic need not be too detailed. For example, when describing communication protocols, it is sufficient to describe the general purpose of protocols together with the names and uses of some contemporary examples (such as TCP/IP, FTP, SMTP, V34 and CSMA/CD). Similar treatment should be afforded to the other characteristics.

Given that networks and internetworks share some characteristics, the major characteristics of internetworks only relate to hardware, communications and functionality. The unique hardware associated with internetworks relates to repeaters, bridges, routers and gateways. The key communication characteristic of internetworks is the adoption of national or global communication standards and protocols (such as Open Systems Interconnection and TCP/IP). The key functional characteristic of internetworks is the transparent way in which communication with remote systems is achieved.

Contemporary examples of each type of network must be described. These descriptions must include the Internet and at least one intranet. Each network must be described in terms of hardware, software, communications, functionality and security. Appropriate networks (in addition to intranets and the Internet) include a range of local-area networks (such as Novel networks, Windows networks, Apple Talk networks, RM-NET) and wide-area networks (such as CompuServe or some corporate networks).

The importance of national and international standards must be emphasised (particularly the latter). For the purposes of this unit specification, the two internetworking standards that candidates should understand are TCP/IP and the OSI reference model. It is desirable (although not essential) for candidates to relate TCP/IP to the OSI model.

Outcome 3

This outcome relates to the candidate's use of a specific network. The general characteristics introduced to the candidate as part of Outcome 2 can now be related to the school or college network system. A wide range of commonly available networks are appropriate, such as RM-NET, Apple Talk, Windows and UNIX systems, together with a large number of public and private networks (such as CompuServe and the Internet). The characteristics of a specific network must be described. The characteristics relate to hardware, software, communication and security. Note that the functional characteristics are covered in PC (b). Some network systems are more capable than others and candidates are only required to describe the actual facilities available on the network system which they normally use. However, it is expected that any network will display all of the characteristics to a greater or lesser extent. The facilities provided by a network can be considered under two headings - services (such as resource sharing, file transfer, e-mail and search services) and resources (such as information and files).

Candidates are required actually to use the defined facilities. At a minimum, candidates would be expected to undertake the following activities:

- log-on
- search for information
- transfer a file
- send an e-mail.
- log-off.

Of course, the capability and complexity of each of these tasks will vary from system to system and centres may need to tailor each task to their own system (such as using e-mail to transfer files if the system does not support a more direct form of file-transfer protocol). In the context of the Internet, this might involve signing on to their Internet Service Provider (log-on), locating a particular Web page using a search engine (search for information), downloading a file from an FTP site (file transfer), sending an e-mail over the Internet (send an e-mail) and breaking the connection to the ISP (log-off).

While there are no national or international standards of network conduct, there has evolved a code of conduct known as network etiquette (or 'netiquette'). This relates to users' conduct when using a computer network and, in addition to general advice on network conduct, it also provides advice on e-mail, file transfer and information browsing. A detailed description of network etiquette will be provided as part of the support materials.

The legal requirements include legislation relating to data protection, copyright and computer misuse; candidates should be familiar with the main regulations as they relate to computer networks. Security requirements vary from system to system but normally include the use of accounts, passwords and access rights.

Outcome 4

This outcome provides candidates with an opportunity to investigate an aspect of computer networking. Appropriate topics could include:

- Search Engines
- Methods of encryption
- HTML or Java
- A Communication protocol eg HTTP, SMTP or FTP
- IP addressing.

The candidate should write a report of 800–1000 words on their chosen topic.

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

The content/context section (above) provides additional guidance on the delivery of the outcomes. This section focuses on learning and teaching.

Candidates will require access to appropriate computer hardware and software at various stages within this unit. To enliven learning, the use of video, audio and multimedia learning aids is recommended. While the distribution of time between the outcomes will vary, candidates may be expected to complete each outcome within the following time scale:

Outcome 1 ..... 8 hours
Outcome 2 ..... 8 hours
Outcome 3 ..... 14 hours
Outcome 4 ..... 10 hours

If this unit is delivered as part of a course, then the course documentation will provide further information on teaching and learning in a course context. This documentation will identify a number of 'themes' to facilitate holistic teaching and learning across the course.

National Unit Specification: support notes (cont)

UNIT Computer Networking (Higher)

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

Centres may use the instruments of assessment which are considered to be most appropriate. Examples of instruments of assessment which could be used are as follows:

Outcome 1	Extended-response questions on the applications and benefits of two local-area networks and two wide-area networks (to include intranets and the Internet).
Outcome 2	Extended-response questions on the characteristics of two computer networks and two internetworks (to include intranets and the Internet).
Outcome 3	• Practical exercise requiring the candidate to use the facilities provided by a specific computer network (PCs (c), (d), (e)). • A short report on the characteristics and facilities provided by an actual network PCs (a) and (b). • An essay covering the candidate's knowledge of network etiquette and security and legal aspects of network use.
Outcome 4	A practical assignment requiring the candidate to investigate an aspect of computer networking.

During the work of the unit, candidates should have several opportunities to develop their practical skills and should be assessed at appropriate points. Terminology should be presented in context throughout the unit. Where the candidate is unsuccessful in achieving an outcome, provision should be made for further work and reassessment.

There is an opportunity to integrate assessment. The extended-response questions for Outcomes 1 and 2 could be combined into a single paper. The other assessment activities may have to be undertaken separately.

Written evidence may take various forms including hand-written and word-processed text or other forms of written communication that are more suited to candidates with physical disabilities. Candidates should not be required to produce a specific form of written text (such as word-processed text) unless this is stipulated within the unit specification or instrument of assessment.

National Unit Specification: support notes (cont)

UNIT Computer Networking (Higher)

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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).

SCOTTISH QUALIFICATIONS AUTHORITY

National Unit Specification: general information

UNIT Multimedia Technology (Higher)

NUMBER D102 12

COURSE Computing (Higher)

SUMMARY

This unit introduces candidates to the software, hardware, media elements and standards which underpin multimedia. Candidates will develop practical competence in the use of hardware and software to capture and process media elements and develop problem-solving abilities in the domain of multimedia technology.

OUTCOMES

1 Demonstrate knowledge and understanding of multimedia technology.
2 Use multimedia technology to capture and present media elements.
3 Apply knowledge and understanding of multimedia technology to develop solutions to problems in multimedia.
4 Investigate an aspect of multimedia technology.

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 experience):

- Multimedia (Int 2) unit
- Computer Systems (H) unit
- Computing course at Intermediate 2 level
- Grade 1 or 2 at Standard Grade in Computing Studies.

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 04

National Unit Specification: general information (cont)

UNIT Multimedia Technology (Higher)

CREDIT VALUE

1 credit at Higher.

CORE SKILLS

There is no automatic certification of core skills or core skills components in this unit.

Additional information about core skills is published *in Automatic Certification of Core Skills in National Qualifications* (SQA, 1999).

National Unit Specification: statement of standards

UNIT Multimedia Technology (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 multimedia technology.

Performance criteria

(a) Significant stages in the development of multimedia are accurately described.
(b) Contemporary standards for multimedia hardware and software are correctly described.
(c) Techniques for capturing and processing media elements are accurately explained.
(d) Characteristics of multimedia software are accurately described.
(e) Multimedia computer system requirements are accurately described.

Note on range for the outcome

Development: hardware; software.
Media elements: text; graphics; sound.
Standards: video; audio
Characteristics: audio processing; image processing; video editing; presentation; authoring.

Evidence requirements

Written or oral evidence that the candidate can demonstrate knowledge and understanding of multimedia technology as detailed in PCs (a) to (e) for all classes in the range.

OUTCOME 2

Use multimedia technology to capture and present media elements.

Performance criteria

(a) Media elements are captured correctly and without loss of fidelity.
(b) Media elements are efficiently manipulated.
(c) Media elements are presented in a coherent and attractive manner.
(d) Hardware and software are used efficiently and effectively.

Note on range for the outcome

Media elements: text; graphics; sound.

Evidence requirements

Performance evidence that the candidate can use multimedia technology to capture and present media elements as detailed in PCs (a) to (d) for all classes in the range.

National Unit Specification: statement of standards (cont)

UNIT Multimedia Technology (Higher)

OUTCOME 3

Apply knowledge and understanding of multimedia technology to develop solutions to problems in multimedia.

Performance criteria

(a) The nature of the problem is correctly identified.
(b) The main features of the solution are accurately described.
(c) Design of solution is consistent with requirements and is clear and complete.
(d) Evaluation of solution is clear, complete and concise.

Note on range for the outcome

There are no specific requirements as to the range of contexts within which the outcomes and performance criteria should be demonstrated. For further guidance on the range of content to be covered, see the support notes.

Evidence requirements

Written evidence that the candidate can apply knowledge and understanding of multimedia technology to develop solutions to problems in multimedia as detailed in PCs (a) to (d).

OUTCOME 4

Investigate an aspect of multimedia technology.

Performance criteria

(a) Strategy for the investigation is clear, complete and achievable.
(b) The investigation is carried out systematically in accordance with strategy.
(c) Reporting is clear, concise, accurate and complete.

Note on range for the outcome

There are no specific requirements as to the range of contexts within which the outcomes and performance criteria should be demonstrated. For further guidance on the range of content to be covered, see the support notes.

Evidence requirements

Written or oral evidence that the candidate can investigate an aspect of multimedia technology as detailed in PCs (a) to (c). This will be in the form of a written report detailing the findings of the investigation into an aspect of multimedia.

National Unit Specification: support notes

UNIT Multimedia Technology (Higher)

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

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

GUIDANCE ON CONTENT AND CONTEXT FOR THIS UNIT

This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Higher level. The details of content given earlier provide information on the delivery and assessment of this unit within a course context.

Corresponding to Outcomes 1–4

This unit is the first in a series of units relating to multi-media and other units are available at Access, Intermediate 1, Intermediate 2 and Advanced Higher levels. However, unlike these other units, this unit specialises in the technological aspects of multimedia systems.

Throughout this specification, 'graphics' is used in a generic sense to encompass original artwork, clipart, photographic images, animations and video. For the purpose of the evidence requirements, only one of these graphic types is required unless more than one type is explicitly stated in the evidence requirements. Treatment of animations and video should be light. Candidates should not be expected to create or capture their own video images nor create their own animations but they should be capable of incorporating video images and animations into a presentation. Issues relating to the copyright of media elements should be emphasised throughout.

Outcome 1

This outcome relates to the hardware and software facilities required to capture and process multimedia data. Coverage should be sufficient to allow candidates to make use of the relevant hardware and software without detailed technical knowledge. There is considerable scope for integration of both the teaching and assessment with Outcome 1. This can be regarded as the theoretical underpinning for the practical application of multimedia which takes place in Outcome 2. Ideally, the delivery and assessment of the two outcomes should be integrated, eg teaching about the theoretical background of sound cards should take place in parallel with the use of sound cards to capture and play audio.

Candidates should be aware of the development of multimedia including the predecessors of multimedia, eg hypertext, interactive video and computer-based training. They should be aware of the development of the hardware components (such as CD ROMs, video capture boards and sound cards) which make multimedia feasible and the software (such as audio and graphics utilities and presentation and authoring packages) which drive it. Candidates should be aware of the hardware and software used to create, capture and manipulate the various media elements, eg text, audio and video (including clipart, original drawings, diagrams, images, graphs and charts, animations and video).

Candidates should be aware of how different hardware components can be used to capture data, manipulate it and output it (eg use of a video capture card to capture video data, a video compression card to compress it and a video display card (such as MPEG) to display it.

Candidates should be aware of how different software facilities may be used to capture, manipulate and display data, eg use of scanner software to capture a graphic, use of a bitmap editor to amend it and use of a presentation graphics package as the final display tool. With regard to text processing, they should be aware of word-processing and desktop publishing software and the differences and similarities between these, as well as the text-processing facilities available in multimedia software. With regard to images, they should be aware of both image-capture hardware, eg scanners, video capture card and image-display hardware, eg graphics cards, video playback cards. Candidates should have a basic knowledge of the limitations imposed by display hardware, eg image size, colour depth, frame rate etc. and be aware of the need for compression of video files. They should also be aware of image-processing software, eg bitmapped graphics (painting) software, vector graphics (drawing) software and image manipulation software and have some exposure to video-editing software. With regard to audio, candidates should be aware of the distinction between sampled and synthesised sound and the use of sound cards in the recording of sampled sounds and the playback of synthesised sounds and the use of MIDI instruments as output devices. Candidates should be aware of the factors affecting the quality of both sampled sounds (sample resolution, frequency) and MIDI sounds (FM vs. wavetable synthesis) and have a basic knowledge of the functions of audio processing software, eg recording, playback, editing, sequencing and compression.

Candidates should be aware of the range of multimedia authoring programs available, from straightforward presentation packages through to sophisticated authoring systems. They should be aware of the different programming models used in authoring systems, eg icon-based models and script-based models.

Multimedia computer system requirements should be described in terms of the currently applicable standards, eg the MPC I, MPC2 and MPC3 standards for PC-based systems.

Outcome 2

This outcome relates to the capture and presentation of multimedia data. There is considerable scope for integration of both the teaching and assessment of this outcome with Outcome 1, as described earlier.

Candidates should be able to manipulate text characteristics such as font, size, colour and alignment. They should be able to manipulate both bit-mapped and vector graphics by scaling, rotating, translating, clipping, panning and filling. Some consideration should be given to conversion between vector and bit-mapped image formats and additional manipulations available on bit-mapped images, eg colour depth, colour palette, blending, smearing, gamma correction etc. In addition, candidates should be able to enter, edit and play MIDI scores through a sound and be able to sample, edit and play back sampled sounds. Editing facilities should include copying, combining and adding special effects, eg echo, fade in/out, pan. Candidates should also be able to capture and edit video.

National Unit Specification: support notes (cont)

UNIT Multimedia Technology (Higher)

Outcome 3

This outcome relates to the application of knowledge and understanding of multimedia technology to develop solutions to problems in multimedia. Possible problem areas might include (but are not confined to): a biography; a current or historical event; a description or review of a film or television program or a musical or dramatic performance; a tourist guide to an area; a computer-assisted learning package on a specific topic.

Candidates should be capable of selecting the media elements that can be used to solve a specified problem and defining the sequence in which these ought to be presented. Note that candidates are only required to design a solution to a problem and evaluate their design - they are not required to implement the solution. However, there may be some scope for integration with Outcome 2, in which case the proposed solution could be implemented to meet the performance criteria for Outcome 2. The evaluation should assess how well the proposed solution meets the problem requirements and give consideration to ways in which it could be improved.

Outcome 4

This outcome relates to the investigation of an aspect of multimedia technology. Suitable topics include (but are not confined to) audio hardware or software, video hardware or software, graphics hardware or software, authoring packages, multimedia and the World Wide Web. Candidates should have access to a range of books and magazines containing information about multimedia technology. Ideally, they should also have access to the World Wide Web, to allow them to search for relevant information. Candidates should draw up a strategy and have it approved by the teacher/lecturer prior to carrying out the investigation.

National Unit Specification: support notes (cont)

UNIT Multimedia Technology (Higher)

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

This unit is one in a series of units relating to multimedia. The content/context section (above) provides additional guidance on the delivery of the outcomes. This section focuses on learning and teaching.

Candidates will require access to a wide range of multimedia hardware and software, eg textprocessing software, sound cards and audio processing software, vector and bit-mapped graphics software, scanners and multimedia software. However, in most cases simultaneous access will not be required and in many instances one example of each type of hardware and software will be sufficient for a class. To enliven learning, the use of video, audio and multimedia learning aids is recommended.

While the distribution of time between the outcomes will vary, candidates may be expected to complete each outcome within the following time scale:

Outcome 1 ..... 8 hours
Outcome 2 ..... 14 hours
Outcome 3 ..... 8 hours
Outcome 4 ..... 10 hours

If this unit is delivered as part of a course (or programme of units), then the course documentation will provide further information on teaching and learning in a course context. This documentation will identify a number of 'themes' to facilitate holistic teaching and learning across the course.

National Unit Specification: support notes (cont)

UNIT Multimedia Technology (Higher)

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

Centres may use the instruments of assessment which are considered to be most appropriate. Examples of instruments of assessment which could be used are as follows:

Outcome 1	A series of short-answer questions requiring candidates to give written descriptions of the development of multimedia, techniques for processing and capturing media elements, characteristics of multimedia software and computer system requirements.
Outcome 2	Two short multimedia presentations requiring candidates to capture and manipulate media elements. Successful completion of these may be recorded on a checklist.
Outcome 3	A brief report by the candidate, giving an analysis of the problem, the design of a proposed solution and an evaluation of the proposed solution.
Outcome 4	A written report on an aspect of multimedia technology (800 - 1000 words).

Advice on generating evidence

During the work of the unit, candidates should have several opportunities to develop their practical skills and should be assessed at appropriate points. Terminology should be presented in context throughout the unit. Where the candidate is unsuccessful in achieving an outcome, provision should be made for further work and re-assessment.

There are opportunities to integrate assessment within this unit. Instead of assessing Outcomes 1 and 2 separately, it would be possible to assess candidates' capability to describe each item of hardware or software at the same time as assessing their ability to use it.

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 alternative outcomes for units. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs/Candidates whose First Language is not English* (SQA, 1998).