SCOTTISH QUALIFICATIONS AUTHORITY

COMPUTING
Advanced Higher

Second edition – published December 1999

SCOTTISH QUALIFICATIONS AUTHORITY

NOTE OF CHANGES TO ADVANCED HIGHER ARRANGEMENTS SECOND EDITION

COURSE TITLE: Computing (Advanced Higher)

COURSE NUMBER: C017 13

National Course Specification

Course Content: No significant amendments

National Unit Specification:

No significant amendments

SCOTTISH QUALIFICATIONS AUTHORITY

National Course Specification

COMPUTING (ADVANCED HIGHER)

COURSE NUMBER C017 13

COURSE STRUCTURE

The course comprises two mandatory units and one optional unit:

Mandatory units

D095 13	***Software Development (AH)***	***0.5 credit (20 hours)***
D098 13	***Computing Project (AH)***	***1.5 credits (60 hours)***

Optional units

One selected from:

D101 13	***Artificial Intelligence (AH)***	***1 credit (40 hours)***
D093 13	***Computer Systems (AH)***	***1 credit (40 hours)***
D103 13	***Data Communications (AH)***	***1 credit (40 hours)***

In common with all courses, this course includes 40 hours over and above the 120 hours for the component units. This is for induction, extending the range of learning and teaching approaches, support, consolidation, integration of learning, and preparation for external assessment. This time is an important element of the course and advice on its use is included in the course details.

RECOMMENDED ENTRY

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

- Computing course at Higher level

Administrative Information

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 02

National Course Specification: general information (cont)

COURSE Computing (Advanced Higher)

CORE SKILLS

Core skills for Advanced Higher remain subject to confirmation and details will be available at a later date.

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

COURSE Computing (Advanced 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.

Computing at Advanced Higher builds upon the knowledge and understanding of computing concepts gained at Higher level, allowing candidates to extend the breadth and depth of their knowledge. Candidates further develop their problem-solving skills in a practical context through undertaking a substantial project which involves the development of a solution to a real computing problem.

Computing at Advanced Higher level allows candidates to make the transition from a structured learning and teaching environment to more independent, flexible learning. Emphasis is placed on developing personal study skills and on encouraging an analytical approach to problem-solving. The course develops practical competencies and an understanding of generic computing concepts which are applicable to a range of contexts and activities. This provides the opportunity for candidates to consolidate and extend their experience and to prepare for further study or for a career involving computing.

The aims of this course are to:

- provide candidates with knowledge and understanding of contemporary computing technology
- provide candidates with an understanding of underlying fundamental computing concepts
- encourage independent investigative enquiry into developments in computing
- develop an analytical approach to problem-solving which will enable candidates to identify and apply methodologies appropriate to the development of computing solutions
- enable candidates to develop skills of analysis, synthesis, evaluation and communication 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 analytical and problem-solving abilities gained through the study of Advanced Higher Computing provide a basis for candidates to have a creative role in a modern technological society.

National Course Specification: course details (cont)

COURSE Computing (Advanced 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 the tools to develop computing solutions to real problems and the opportunity to design, implement and evaluate a substantial computing project. The third optional unit builds into the course the flexibility to be responsive to the needs, interests and aspirations of the candidate. By studying the component units within the context of the Advanced 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 candidates' 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).

Unit contents

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

National Course Specification: course details (cont)

COURSE Computing (Advanced Higher)

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. 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 to attain 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 exemplar 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 Advanced Higher level.

	Software Dev.		**Computing Project**			**Data Comms**		
Assessment	1	2	1	2	3	1	2	3
1	✓			✓				
2		✓	✓		✓			
3						✓	✓	
4								✓

Table 1 - Assessment matrix for Computing at Advanced Higher level

It assumes that the optional unit is Data Communications; similar tables will be produced for alternative options. The table shows how the eight outcomes within the component units can be assessed using four assessment activities. Outline details of each instrument of assessment follow.

Assessment 1

This assessment combines the knowledge and understanding gained as part of the Software Development unit with the new knowledge and understanding gained as part of the Computing Project unit. It assumes that the candidate has selected a software development task as the basis of his/her project. This would permit the knowledge of the systems development gained through the Software Development unit (Outcome 1) to be extended to encompass additional knowledge and understanding required to undertake the Project unit (Outcome 2).

Assessment 2

This assessment combines the assessment of an outcome within the Software Development unit with two outcomes from the Computing Project unit. It assumes that the project relates to a software development task and that candidates will take the software produced as part of the Software Development unit (Outcome 2) and use this as the basis of their project (Computing Project, Outcomes 1 and 3).

Assessments 3 and 4

Details of these assessments are contained within the appropriate unit specification.

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.

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 Advanced Higher level will consist of two components with weightings as follows:

- coursework 50%
- written examination 50%

Coursework

The coursework component is designed to assess candidates in their knowledge and skills of applied computing – analysis, design, implementation and evaluation – as they have been developed across component units. The Computing Project unit integrates a wide range of such knowledge and skills and the assessment for this unit – a project – contributes significantly to the coursework component. By completing a project, candidates will provide evidence of analysis, design and implementation. The coursework component requires that in addition candidates complete a written evaluation of their project. The marking guidelines for the unit assessment include advice on marking the evaluation to complete the coursework element.

The project will be subject to moderation both for unit and course purposes.

Written examination

The written examination will comprise a single paper of 2 hours' duration. The paper will consist of a mandatory section (30%) and an optional section (70%) comprising questions for each of the optional units. Candidates will be provided with a choice of question within each section. The written examination will include extended-response questions which allow candidates the opportunity to demonstrate the ability to: integrate knowledge, understanding and problem-solving skills acquired in the component units and apply an analytical approach to problem-solving in more complex or less familiar contexts.

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).

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*.

National Course Specification: course details (cont)

COURSE Computing (Advanced Higher)

Grade C	Grade A
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 that integrates outcomes from more than one unit *and skills are comfortably transferred and synthesised.*
Application of knowledge, understanding and skills in more complex contexts.	
Candidates are able to apply their knowledge and understanding in straight-forward 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 artefact that results from practical activity meets the standards defined by the associated performance criteria.	The artefact 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 exhibits 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*.

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 competencies 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 appears 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 Advanced Higher course.

National Course Specification: course details (cont)

COURSE Computing (Advanced Higher)

Integrative topic	Software Development	Computing Project	Artificial Intelligence	Computer Systems	Data Comms
Principles		✓		✓	✓
Practical abilities	✓	✓		✓	✓
Applications			✓		✓
Software development	✓			✓	
Problem-solving	✓	✓	✓	✓	
Objects and operations	✓	✓	✓	✓	

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

The above table illustrates that practical skills are a part of several units 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 Computing Project 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 exemplar 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 and some unit specifications provide specific URLs where relevant resources can be found. An important part of candidates' learning is the acquisition and development of a technical vocabulary and a glossary of terms will be produced to define the meaning of the terms used within this documentation.

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 that will give additional information on the delivery of Computing courses and units.

National Course Specification: course details (cont)

COURSE Computing (Advanced Higher)

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
- preparation of practical coursework for course 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 Software Development (Advanced Higher)

NUMBER D095 13

COURSE Computing (Advanced Higher)

SUMMARY

This unit is designed to develop knowledge and understanding of systems analysis, to extend knowledge and understanding of the software development process and to provide the opportunity to apply a methodology to analyse and design a software solution to a problem.

OUTCOMES

1 Explain aspects of the software development process in relation to systems analysis and design.
2 Apply a methodology to analyse a problem and design a software solution.

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 Higher level

CREDIT VALUE

0.5 credit at Advanced Higher.

Administrative Information

Superclass: CB

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 02

National Unit Specification: general information (cont)

UNIT Software Development (Advanced Higher)

CORE SKILLS

Core skills for Advanced Higher remain subject to confirmation and details will be available at a later date.

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 (Advanced Higher)

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

OUTCOME 1

Explain aspects of the software development process in relation to systems analysis and design.

Performance criteria

(a) The purposes of systems analysis and design are explained accurately.
(b) The stages of the software development process are explained accurately.
(c) Roles and relationships between personnel are described correctly.

Note on range for the outcome

Purposes: feasibility study; system investigation; operational requirements specification.
Stages: analysis of operational requirements; system specification; software design and validation; implementation and testing of components; integration and testing of whole system; documentation and maintenance.
Personnel: management; project leader; systems analyst; programming team.

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 classes in the range.

OUTCOME 2

Apply a methodology to analyse a problem and design a software solution.

Performance criteria

(a) Methodology is used correctly to analyse a problem and produce a system specification.
(b) Methodology is used correctly to design a software solution from a system specification.
(c) Validation of a software design is 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

Performance evidence that the candidate can apply a methodology to analyse a problem and design a software solution as detailed in PCs (a) to (c).

National Unit Specification: support notes

UNIT Software Development (Advanced 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 Advanced 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 and 2

This unit is the last in a series of units relating to Software Development and other units are available at Intermediate 1, Intermediate 2 and Higher levels.

Outcome 1

In this outcome, candidates are required to explain aspects of the software development process in relation to systems analysis and design. The candidate is expected to explain the purposes of systems analysis and design (PC (a)) and explain the stages of the software development process (PC (b)). In addition, the roles and relationships between personnel should be described correctly (PC (c)). A description of each performance criterion follows.

a) The purposes of systems analysis and design are explained accurately.

The purposes in the range are defined as:

- feasibility study
- system investigation
- operational requirements specification

The candidate should gain an insight into the purposes of systems analysis and design and in particular be familiar with the documentation produced as a result of this process. In delivery, good use can be made of case studies. Candidates may be allowed to develop their own feasibility study, system investigation and operational requirements specification for a given small-scale problem. The feasibility study is used to determine whether the objectives of the problem definition are possible, before proceeding. The system investigation considers what must be done to solve the problem. The operational requirements specification describes what the system must do and marks the end of the logical design phase.

b) The stages of the software development process are explained accurately.

The stages defined in the range are:

- analysis of operational requirements
- system specification
- software design and validation

- implementation and testing of components
- integration and testing of whole system
- documentation and maintenance

Software development forms part of the whole systems' life cycle and is likely to proceed after the systems analysis and design. Real-life case studies would provide a suitable and meaningful context in which the required knowledge can be gained. The methodical nature of the software development process should be stressed. The beginning of the software development process is marked by the analysis of the operational requirements in order to determine fully what the software must do. The resultant document, the system specification, fully describes what the software has to do in terms of its functions and performance. The design phase of the software development process will then proceed. The design will be based on the system specification. The design will be validated as it proceeds. Implementation and testing of software modules developed from the design documentation will take place before they are integrated and futher testing takes place. It is important for the candidate to appreciate that both user and system documentation will take place at each stage. The requirement for software maintenance is always considered.

c) Roles and relationships between personnel are described correctly.

The personnel defined in the range are:

- management
- project leader
- systems analyst
- programming team

The roles and the relationships between the personnel involved in systems analysis, design and software development should be described.

Outcome 2

In this outcome, candidates are required to apply a methodology to analyse and design a software solution to a problem. This will require the candidate to analyse the operational requirements of a problem (PC (a)). The solution can then be designed using an appropriate methodology (PC (b)). The candidate will also be required to validate a software design (PC (c)). A description of each performance criterion follows.

a) Methodology is used correctly to analyse a problem and produce a system specification.

The analysis of the operational requirements to produce the system specification marks the start of the software development phase. Candidates should analyse an operational requirements specification in order to produce a system specification. Given that they may have already produced an operational requirements specification, this document could then be analysed to produce a system specification. An alternative approach would be to allow the candidate to be given an existing feasibility study, system investigation and operational requirements in order to produce a system specification. Athough this may reflect real-life practices, it is unlikely that time would permit this except in exceptional cases.

b) Methodology is used correctly to design a software solution from a system specification.

Having obtained the system specification, the candidate should then use this to design a software solution to the problem. Candidates should employ an appropriate design methodology in carrying out this activity. The design should be carried out in such a way that there is a clear indication of division into modules and the data flow between modules. The design should be to a level appropriate for implementation.

c) Validation of a software design is accurate and complete.

The candidate is required to validate a software design. This may be for a small-scale software design or, if time permits, for the candidate's own software designs produced above. In validating the design, candidates should employ a dry run or walkthrough of the design by working sets of test data through the design. The test data should cover all possibilities.

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

This is the last in a series of units relating to Software Development. 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

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.

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 aspects of the software development process in relation to systems analysis and design.
Outcome 2	An assignment on the analysis and design of the solution to a problem.

National Unit Specification: support notes (cont)

UNIT Software Development (Advanced Higher)

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.

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 Computing Project (Advanced Higher)

NUMBER D098 13

COURSE Computing (Advanced Higher)

SUMMARY

This unit develops the ability of the candidate to analyse a computing problem and to design and implement a computer-based solution.

OUTCOMES

1 Demonstrate an analytical approach to a computing problem.
2 Demonstrate knowledge and understanding of relevant computing concepts.
3 Implement a solution to a problem.

RECOMMENDED ENTRY

While entry is at the discretion of the centre, candidates would normally be expected to have attained the Computing course at Higher level (or equivalent experience).

CREDIT VALUE

1.5 credits at Advanced Higher.

Administrative Information

Superclass: CB

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 02

National Unit Specification: general information (cont)

UNIT Computing Project (Advanced Higher)

CORE SKILLS

Core skills for Advanced Higher remain subject to confirmation and details will be available at a later date.

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 Computing Project (Advanced Higher)

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

OUTCOME 1

Demonstrate an analytical approach to a computing problem.

Performance criteria

(a) Main stages of the analytical approach are accurately identified.
(b) Project proposal is clear and concise.
(c) Project specification is clear and accurate.
(d) Selected strategy is appropriate to the nature of the problem.
(e) Design of the solution is clear 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 plan a project as specified in PCs (a) to (e).

OUTCOME 2

Demonstrate knowledge and understanding of relevant computing concepts.

Performance criteria

(a) Prerequisite knowledge and skills are accurately identified.
(b) Sources of information are accurately identified.
(c) Information is collated efficiently.
(d) Knowledge and understanding are accurately communicated.

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 possesses relevant knowledge and understanding as detailed in PCs (a) to (d).

National Unit Specification: statement of standards (cont)

UNIT Computing Project (Advanced Higher)

OUTCOME 3

Implement a solution to a problem.

Performance criteria

(a) Hardware and software are used efficiently and effectively.
(b) Implemented solution adheres to the design.
(c) The solution is accurately documented using appropriate vocabulary.
(d) The solution is systematically tested using appropriate test data.

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

Performance evidence that the candidate can implement a solution as detailed in PCs (a) to (d). In addition to observation evidence, evidence in the form of an artefact (such as a software product) will also be provided.

National Unit Specification: support notes

UNIT Computing Project (Advanced 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 Advanced 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 to 3

This unit is designed to integrate a wide range of knowledge and skills that candidates will acquire through undertaking a number of computing-related courses and units. It is expected that most candidates will have undertaken the Computing course at Higher level (or equivalent programme of units) prior to undertaking this unit.

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

A number of project specifications will be produced. Each specification will relate to a complex problem which can be resolved through the application of the candidate's knowledge and understanding of computing. Potential problem areas include software development and a wide range of developments relating to computer systems (such as the Internet and multimedia). Further guidance on the conduct of the project will be provided within the project specification.

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.

Given the level and recommended entry requirements of this unit, it is expected that candidates will be able to research and implement solutions without a great deal of teacher/lecturer assistance. Candidates should possess a wide range of knowledge and skills prior to undertaking this unit. It is anticipated that the teacher/lecturer will perform as mentor to the candidate during the project, guiding and advising the candidate when appropriate.

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 ..... 15 hours
Outcome 2 ..... 15 hours
Outcome 3 ..... 30 hours

National Unit Specification: support notes (cont)

UNIT Computing Project (Advanced Higher)

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.

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:

Outcomes 1-3 All outcomes will be used through project activity. The project will be structured so that it generates all of the prescribed 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.

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 (Advanced Higher)

NUMBER D101 13

COURSE Computing (Advanced Higher)

SUMMARY

This unit is designed to develop knowledge and understanding of knowledge domains, search methods and expert system shells. Problem-solving skills, in the context of artificial intelligence, are extended as are practical abilities in the use of specialised software, in particular expert system shells.

OUTCOMES

1 Analyse the processes related to problem-solving in artificial intelligence.
2 Analyse the representation of knowledge domains and methods of searching these.
3 Critically appraise features of expert system shells.

RECOMMENDED ENTRY

While entry is at the discretion of the centre, candidates would be most likely to benefit from undertaking this unit if they had gained one of the following qualifications (or possessed equivalent experience):

- Artificial Intelligence (H) unit
- Computing course at Higher level
- Higher Grade Computing

Administrative Information

Superclass: CB

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 02

National Unit Specification: general information (cont)

UNIT Artificial Intelligence (Advanced Higher)

CREDIT VALUE

1 credit at Advanced Higher.

CORE SKILLS

Core skills for Advanced Higher remain subject to confirmation and details will be available at a later date.

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 (Advanced Higher)

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

OUTCOME 1

Analyse the processes related to problem-solving in artificial intelligence.

Performance criteria

(a) Processes are accurately described.
(b) Problems are systematically solved using an appropriate process.
(c) Evaluation of solution is complete and concise and relates to appropriate criteria.

Note on range for the outcome

Processes: problem abstraction; symbolic representation; heuristics.

Evidence requirements

Written or oral evidence that the candidate can analyse the processes related to problem-solving in artificial intelligence in accordance with PC (a). This may be evidenced by a series of short-response questions. Performance evidence that the candidate can use an appropriate process to solve a specific problem systematically in accordance with PC (b).

Written or oral evidence that the candidate has evaluated the solution to the problem in accordance with PC (c).

OUTCOME 2

Analyse the representation of knowledge domains and methods of searching these.

Performance criteria

(a) Knowledge domains are accurately and efficiently represented.
(b) Search techniques are accurately explained and compared in relation to efficiency and the effect of combinatorial explosion.
(c) Applications of search techniques are accurately explained.
(d) Search techniques are accurately applied to solve simple problems.

Note on range for the outcome

Knowledge domains: search trees; logic; semantic nets; frames.
Search techniques: exhaustive searching; heuristic searching.
Applications: natural language processing; vision; robotics.

Evidence requirements

Written or oral evidence that the candidate can analyse the representation of knowledge domains and methods of searching these in accordance with PCs (a) to (c). This may be evidenced by completing a series of extended-response questions. Performance evidence is required that the candidate can represent a knowledge domain accurately and efficiently (PC (a)) and solve a simple problem by applying an appropriate search technique in accordance with PC (d).

OUTCOME 3

Critically appraise the features of expert system shells.

Performance criteria

(a) Features of expert system shells are accurately described.
(b) Implementation of expert systems using different shells is compared in relation to appropriate criteria.

Note on range for the outcome

Expert system shells: not supporting uncertainty; supporting uncertainty.
Features: search techniques; justification facilities; method of building the knowledge base.
Criteria: ease of building the knowledge base; quality of user interface; quality of justification facilities; effect of different search techniques; the effect of uncertainty.

Evidence requirements

Written or oral evidence that the candidate can critically appraise expert system shells in accordance with PCs (a) and (b) for all classes in the range. At least two shells must be appraised (as detailed in the range statement) and the features of each described and compared against the defined criteria.

National Unit Specification: support notes

UNIT Artificial Intelligence (Advanced 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

The details of content given earlier provide information on the delivery and assessment of this unit within a course context. This unit may be delivered as a stand-alone unit or in combination with other units as part of the Computing course at Advanced 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 in the unit Artificial Intelligence (H), which is one of the optional units in the Computing course at Higher level.

It is anticipated that this unit will be undertaken in a variety of contexts by a wide range of candidates. Previous knowledge and experience of artificial intelligence is desirable and it is strongly recommended that this unit is viewed as a continuation of, or an extension to, the unit Artificial Intelligence (H). However, it might be possible for highly motivated candidates with wide-ranging knowledge and experience of other computing topics to benefit from undertaking this unit without prior knowledge of artificial intelligence.

Outcome 1

This outcome is about analysing some of the processes related to problem-solving in artificial intelligence. The particular processes about which candidates are required to develop knowledge and understanding are problem abstraction, symbolic representation and heuristics. Candidates must also use these processes to solve problems.

Candidates are required to describe each of the processes accurately (PC (a)) and to solve problems systematically using an appropriate process (PC (b)). Due to the interdependence of these performance criteria, it is suggested that both performance criteria are considered together. It is recommended that at first existing solutions to simple problems be analysed and then the techniques learned are applied to other simple problems. It is necessary that candidates develop an understanding of how humans solve such problems before similar problems are solved on a computer. From an analysis of solutions to even the simplest problem it will become apparent that most solutions are not trivial.

Candidates should become aware of the following:

- In many cases, people use heuristics to solve problems. This means that they bring their past experience and knowledge gained in similar situations to the current problem. A good heuristic should reduce the time needed to find a solution
- Before a problem can be solved, it must be clearly understood. This means that the starting state and final goal must be known. Any constraints on the solution have to be identified. This is known as problem abstraction

- Symbolic representation is a useful way of illustrating the state space. Representing the starting state, the final goal and valid intermediate states in a diagram is a clear and unambiguous way to solve a problem
- Especially when problems are to be solved using a computer, it is important to be able to identify the essential elements and then to be able to represent these in a clear and unambiguous way

Classic problems such as those involving water jugs, sliding blocks, cannibals and missionaries, or the fox, goose and bag of grain, along with traditional board games and mathematically-based brainteasers would provide suitable contexts for analysing the required processes.

The solution should be evaluated (PC (c)). The evaluation must be complete, concise and related to appropriate criteria that at this level candidates should be expected to select based on their prior experience. For example, in the unit Artificial Intelligence (H), candidates are directed to 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?

Similar evaluation requirements occur in other units. The criteria chosen by the candidates should be appropriate to the context of the particular problem being solved.

Outcome 2

This outcome relates to the representation of knowledge domains and methods of searching these. Candidates are required to represent domains of knowledge accurately and efficiently (PC (a)). The methods considered must include search trees, logic, semantic nets and frames. Candidates should already be familiar with state space diagrams so these would provide a suitable introduction to the use of search trees. When representing knowledge using logic, it would be appropriate to include practical work using Prolog, or a similar programming language. At this level, it is not necessary for candidates to become familiar with the standard logic symbols used in many of the textbooks: it is quite acceptable for words like ‘if’, ‘and’, ‘or’, ‘for all’ to be written out in full. The analysis of semantic nets should lead to candidates understanding the concept of inheritance. That is, when knowledge can be classified into groups, the objects in a group inherit all the characteristics associated with the group. Again Prolog, can be used, with facts representing the basic information and rules representing the inherited information. The use of frames which mimic the broad conceptual structures of human brains (where relevant facts relating to pre-existing knowledge are assumed and additional data stored in appropriate slots) is another method of knowledge representation which candidates must analyse. Again, consideration of these could include practical work using Prolog.

National Unit Specification: support notes (cont)

UNIT Artificial Intelligence (Advanced Higher)

After learning how artificial intelligence represents knowledge, candidates should consider ways in which the knowledge base can be searched. The purpose of the search is to solve problems in a logical and efficient way. The different search techniques must be explained accurately by the candidate and compared in relation to appropriate criteria (PC (b)). Candidates must also be able to apply search techniques to solve simple problems (PC (d)). Due to the interdependence of these performance criteria it is suggested that both performance criteria, are considered together. Some of the problems considered in Outcome 1 would provide a suitable basis for comparing the different search techniques, as would some of the methods of representing knowledge domains considered for Outcome 2 (PC (a)). Another suitable context would be to compare a route between two locations using exhaustive search and heuristic search techniques. It is essential to ensure that the problems chosen are sufficiently restricted so that undue complexity is not introduced. For example, the tower of Hanoi problem should be simplified by restricting the number of disks to two or three, rather than the standard seven disks.

The search techniques considered must include heuristic searching and exhaustive searching, both breadth-first and depth-first. The relative advantages and disadvantages of these techniques should be compared with particular reference to efficiency and the effect of the combinatorial explosion.

Candidates are required to explain the use of search techniques in three applications (PC (c)). The applications must include natural language processing, computer vision and robotics. It is suggested that candidates do this by investigating a simple problem in each area and, in particular, consider the search methods involved. Suitable topics include pattern matching, parsing, fingerprint analysis and image processing, intelligent robots. In all areas, it is important that candidates are aware that it is the software and not the hardware (robot, camera, etc) which contains the intelligence.

Outcome 3

In this outcome emphasis is on the critical appraisal of a specific aspect of artificial intelligence, expert systems. Expert systems have been relatively successful in capturing human expertise and applying it to a range of problems in medical diagnosis, financial advice and product design.

Outcome 3 provides opportunity for extensive practical work involving the use of expert system shells. In this outcome, the problems tackled should cover more real-life problems than the games and other examples looked at in Outcomes 1 and 2. Throughout this outcome, two expert system shells, one supporting uncertainty and one which does not, should be used. It is fundamentally important that candidates understand what is meant by (un)certainty. Some expert systems now take account of human ‘fuzzy’ logic and, depending on the expert system being used, it may be appropriate for candidates to investigate this concept.

Candidates must demonstrate that they can describe the features of expert system shells accurately (PC (a)). It is recommended that the candidates start by examining and comparing the sample files which are supplied with the expert system shells. Once candidates have a clear idea about what is meant by an expert system shell and about methods of dealing with uncertainty, they should be ready to consider the search techniques used to reach solutions. From Outcome 2, candidates should be familiar with breadth-first and depth-first methods of exhaustive search. Related to expert systems, candidates should also be aware that the inference engine implements either forward or backward chaining. Further comparison of the shells should consider the justification facilities provided. These are part of the user interface and most expert system shells provide two types of justification facilities (How? and Why?). Candidates must also analyse how the knowledge base is built. They should find out about how knowledge is structured and entered into the two available expert systems. Most expert system shells require knowledge to be represented and entered as rules. Some expert systems allow frames or other knowledge representation to be used.

Candidates must demonstrate that they can compare the implementation of expert systems using different shells (PC (b)). At this stage, candidates should assume the role of a knowledge engineer and have opportunity to develop their practical experience in designing and implementing some simple expert systems. Suitable topics might include:

- an expert system which gives advice, for example, an expert system which advises a friend about a suitable sport to take up, or an expert system which advises on which make and/or model of car should be purchased
- a fault diagnosis system, for example, suggesting possible parts which should be repaired or replaced in, faulty gas fire, printer or car engine
- a training system, for example, first aid or theory related to the written driving test
- classifications, for example, classifying different types of vehicles (car, lorry, van, bicycle, etc) based on their characteristics

Candidates must be alerted to the need to restrict the domain of knowledge. For example, in first aid the treatment of burns would provide a sufficiently limited domain of knowledge.

Each topic chosen should be entered into both of the available expert systems and the two implementations compared with regard to the following criteria: the ease of building or entering the knowledge base, the quality of the user interface including the justification facilities, the use of different search techniques (if applicable) and the effect of (un)certainty.

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…….8 hours
Outcome 2…….15 hours
Outcome 3…….17 hours

A range of software will be required to deliver this unit. For Outcome 3, it is essential that candidates have access to two expert system shells, one supporting uncertainty, the other not. If only one package, supporting uncertainty, is available, then it might be possible for candidates to implement the expert system once using the uncertainty option and once ignoring it. However this would not be ideal, as candidates would not benefit from using two different packages and there would be less to consider in the subsequent comparisons of the two implementations.

Wherever possible, practical work should be undertaken so that candidates' experience is not restricted to theoretical aspects. The use of Prolog, or a similar programming language, would enhance much of the work for Outcome 2, by making the link between processes related to problem-solving (albeit at this level, for only very simple problems) and possible computer-based solutions explicit.

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 the last decade 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, most expert system shells require knowledge to be represented and entered as rules while some expert systems allow frames or other knowledge representations to be used. Candidates should see the connection between the work on expert systems for Outcome 3 and what was learned earlier in Outcome 2 (PC (a)). Both Outcomes 2 and 3 also include search techniques. Another example of integrative work is the way in which a problem considered for Outcome 1 could be developed further for Outcome 2 by applying search techniques to obtain the solution.

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 (IA) that could be used are as follows:

Outcome 1

IA1: Written assignment (short-response questions), requiring accurate descriptions of each of the three required processes.

IA2: Written assignment (extended-response question), where the candidate provides a solution to a problem which must be distinct from others considered during earlier studies for this outcome. The solution should be evaluated.

National Unit Specification: support notes (cont)

UNIT Artificial Intelligence (Advanced Higher)

Outcome 2	IA1: Written assignment (extended-response questions), which illustrates different methods of representing knowledge. IA2: Written assignment (extended-response questions), accurately explaining search techniques and comparing them. IA3: Written assignment (extended-response questions) requiring accurate explanations of search techniques uses in three applications. IA4: A practical assignment where the candidate analyses a simple problem, represents the knowledge domain accurately and efficiently, details limitations and assumptions, identifies an appropriate search technique and applies it to produce a solution.
Outcome 3	IA1: Written assignment (extended-response questions), requiring accurate descriptions of the main features of expert systems. IA2: A practical assignment where the candidate develops expert systems which represent a limited domain of knowledge. The subject matter may or may not be familiar to the candidate as all the necessary knowledge should be provided. One version should be implemented on an expert system shell which supports uncertainty and one on an expert system shell which does not. The candidates should be required to compare the two solutions in relation to appropriate criteria.

During the work of the unit candidates should have repeated opportunities, in the context of artificial intelligence, to analyse processes related to problem solving, the representation of knowledge domains and methods of searching these. Practical work, using expert system shells will lead to critical appraisal of two expert system shells. 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, a suitable choice of problem would allow the assessment of some of the performance criteria for Outcomes 1 and 2 to be combined. Similarly, when implementing an expert system for Outcome 3, if candidates gave sufficient detail about the search techniques, this would provide some of the assessment evidence necessary for Outcome 2.

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 Artificial Intelligence (Advanced 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 Systems (Advanced Higher)

NUMBER D093 13

COURSE Computing (Advanced Higher)

SUMMARY

This unit extends the candidate's knowledge of computer organisation and develops a more detailed understanding of computer architecture.

OUTCOMES

1 Describe the architecture of microprocessors.
2 Apply knowledge and understanding of a microprocessor's architecture in the context of assembly language programming.
3 Compare the features of different operating systems.

RECOMMENDED ENTRY

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

- Computer Systems (H) unit
- Computing course at Higher level
- Information Systems course at Advanced Higher level

CREDIT VALUE

1 credit at Advanced Higher.

Administrative Information

Superclass: CD

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 02

National Unit Specification: general information (cont)

UNIT Computer Systems (Advanced Higher)

CORE SKILLS

Core skills for Advanced Higher remain subject to confirmation and details will be available at a later date.

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 (Advanced Higher)

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

OUTCOME 1

Describe the architecture of microprocessors.

Performance criteria

(a) The general architecture of a microprocessor unit is correctly represented in a block structure diagram and this is related to the particular architecture of a specific microprocessor.
(b) The relationship between a microprocessor and its instruction set and addressing modes is correctly explained in the context of a specific microprocessor.
(c) The processor/memory interface for a specific microprocessor is correctly described.
(d) The factors affecting microprocessor performance are accurately explained.
(e) The historical development of microprocessor systems is correctly described.

Note on range for the outcome

Microprocessor architecture: control unit; arithmetic and logic unit; registers; memory interface.

Evidence requirements

Written or oral evidence that the candidate can describe the architecture of a microprocessor as detailed in PCs (a) to (e) for all classes in the range.

National Unit Specification: statement of standards (cont)

UNIT Computer Systems (Advanced Higher)

OUTCOME 2

Apply knowledge and understanding of a microprocessor's architecture in the context of assembly language programming.

Performance criteria

(a) The general relationship between a microprocessor's instruction set and assembly language is correctly explained.
(b) The characteristics of assembly languages are correctly described and accurately related to a specific assembly language.
(c) A range of low-level programming techniques is efficiently and effectively implemented in a specific assembly language.
(d) The assembly language translation process is efficiently and effectively applied.

Note on range for the outcome

Programming techniques: register transfers; branches; comparisons; addressing; stack manipulation.
Translation process: compilation; linking; loading.

Evidence requirements

Written or oral evidence that the candidate can provide the descriptions and explanations as detailed in PCs (a) and (b).

Performance evidence that the candidate can produce assembly language programs as detailed in PCs (c) and (d). At least one program segment representing the implementation of each programming technique must be provided.

OUTCOME 3

Compare the features of different operating systems.

Performance criteria

(a) The generic features of operating systems are accurately described.
(b) The features of a specific operating system are described in relation to these generic features.
(c) The features of two specific operating systems are contrasted and compared.
(d) The features of a specific operating system are used efficiently and effectively.

Note on range for the outcome

Features: memory management; processing modes; input/output; filing system; resource sharing; user interface; applications support; security.

Evidence requirements

Written or oral evidence that the candidate can describe the features of operating systems as detailed in PCs (a) to (c) for all classes in the range. At least two operating systems must be described and these operating systems should exhibit diverse features.

Performance evidence that the candidate can use the features of an operating system as detailed in PC (d) for all classes in the range.

National Unit Specification: support notes

UNIT Computer Systems (Advanced 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 Advanced 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 to 3

This unit is the last in a series of units relating to Computer Systems and other units are available at Access, Intermediate 1, Intermediate 2 and Higher levels. As such, candidates should possess well-developed 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. Earlier units in this series will provide candidates with a broad range of knowledge and skills in the organisation of computer systems. This units aims to deepen their knowledge and understanding of specific aspects of computer systems. Previous units focused on the *organisation* of computer systems; this unit focuses on their *architecture,* which involves a lower-level examination of the operation and design of computer systems.

Outcome 1

This outcome relates to the architecture of a microprocessor unit (MPU). Candidates are required to describe the general architecture of an MPU and relate this to a specific microprocessor (PC (a)). For example, the general architecture of an MPU could be related to the actual architecture of the Motorola 6502 or the Intel Pentium microprocessor; the description would use the register names associated with that actual MPU rather than generic register titles.

Candidates should be aware that each MPU has a unique instruction set (and associated addressing modes) and these should be described for a specific MPU (PC (b)). It would be desirable to compare a simple instruction set (like the 6502) with a more complex instruction set (like the Pentium MMX). Most MPUs have a range of addressing modes which normally include:

- immediate
- indexed
- relative
- indirect

The processor/memory interface must also be explained (PC (c)). In the context of the 6502 MPU, this would describe how the 6502 is connected to its semiconductor memory (normally 6116 CMOS RAM) via its address and data buses.

Candidates will require some knowledge of logic design and pin-outs to describe this interface, although timing diagrams are beyond the scope of this unit.

Candidates must be aware of the factors that affect microprocessor (PC (d)). The main factors are:

- cycle time
- word length
- architecture
- instruction set

The discussion on instruction sets would encompass complex instruction set computers (CISC) and reduced instruction set computers (RISC), as well as fixed and microprogrammed control units. Candidates should be exposed to the various means of measuring MPU performance, including the traditional MIPs rating system.

Candidates should also be aware of the historical development of microprocessor systems (PC (e)). At the time of writing, the most significant MPUs include:

- Intel 4004
- Zilog Z80
- Motorola 6502
- Intel 8088
- Motorola 68000 series
- ARM2/3/250/600 series
- Intel 286/386/486 series
- Intel Pentium/Pro/MMX series

Each MPU should be described in terms of its main design characteristics:

- architecture
- instruction set
- performance

Outcome 2

This outcome relates to assembly language programming. Candidates are required to apply their knowledge of a specific MPU's architecture to produce a series of low-level programs. Candidates are required to know the 1:1 relationship between an MPU's instruction set and the corresponding assembly language (PC (a)). They must also describe the general characteristics of any assembly language and relate this to a specific assembly language (PC (b)). The general characteristics of an assembly language are its:

- instruction set in symbolic form
- use of labels
- use of sub-routines
- macro instructions
- translation process (including linking)

These general characteristics must be related to a specific assembly language. So, for example, candidates might describe how MASM (Microsoft's Macro Assembler) implements each of these facilities.

Candidates are required to use an assembly language to produce small program segments to perform prescribed tasks (PC(c)). These tasks are:

- register transfers
- branches
- comparisons
- addressing
- stack manipulation

Of course, a single program may encompass more than one of these tasks so it may be that two or three programs can cover all of the prescribed programming techniques. However, it is not advised that candidates produce a single large program which encompasses all of these techniques since the degree of complexity is likely to be greater than intended.

Candidates are required to use an assembler to translate their programs (PC (d)). This would typically involve the translation of the source code into intermediate code and the linking of this code with external routines to produce object code. The complexity of this process varies from assembler to assembler and centres should try to use development environments that are typical of contemporary translation tools.

Outcome 3

This outcome relates to the comparative features of contemporary operating systems (OS). For the purpose of evidence requirements, candidates are required to describe the characteristics of at least two diverse operating systems. The systems should be different in at least two or three respects, such as user interface and processing modes. Suitable examples include MS-DOS and Windows or Mac OS and BBC DFS.

The generic features of operating systems must be described (PC (a)). The features are:

- memory management
- processing modes
- input/output
- filing system
- resource sharing
- user interface
- applications support
- security

Candidates are required to relate these features to a specific OS (PC (b)). Most contemporary operating systems would be suitable, such as MS-DOS, Windows and Mac OS. The following descriptions relate to Windows.

Memory management involves the ways in which the operating system utilises the (physical) memory (including garbage collection). Candidates would be required to appreciate the difference between different types of memory (conventional, upper, high, extended, expanded and virtual) and understand the memory model employed by Windows. The processing modes relate to the different ways that an operating system can execute programs and would encompass Windows' ability to multi-task between applications and multi-thread within applications. The description of I/O would relate to the ways in which the operating system can control peripherals and communicate with external systems. This description will also relate to the operating systems use of standards for controlling peripherals and communications purposes. In the context of Windows, this description would encompass the 32-bit driver model, plug-and-play support and standards such as TAPI, MPEG and graphics support. The communication capabilities include built-in support for a range of protocols (such as TCP/IP and IPX) and the inclusion of an Exchange client that includes MAPI support. The filing system relates to the (physical) way in which files are stored on secondary storage devices and the way in which this is represented to the user. In the context of Windows, this would involve a discussion on the file allocation table (FAT32) and its hierarchical system of folders and files. Windows provides simple resource sharing in the form of built-in peer-to-peer networking which allows users to share files and printers. The user interface relates to the way that people communicate with the computer. Windows provides a WIMP environment and a graphical user interface. Candidates would be also be expected to know about the customisation and automation options available within an operating system. Applications support relates to the operating system's facilities for application software and includes such things as software compatibility (ie the range of software available for that platform), support for objects (or 'applets' in Windows) and technical aspects such as object linking and embedding and dynamic data exchange. Some operating systems are more secure than others and candidates are required to describe the security facilities within one specific OS. Single-user operating systems (such as Windows) tend to have limited security features. However, Windows supports basic file security in the form of file attributes and built-in backup software and more sophisticated security for peer-to-peer and client-server configurations.

Each operating system will differ in the extent to which it provides these features and this realisation is a crucial aspect of this outcome. Candidates are required to compare these facilities as they are implemented in two operating systems (PC (c)). A wide range of operating systems could be explored but the selected systems should not be too similar in terms of these features. Suitable operating systems include MS-DOS and Windows, and BBC DFS and Mac OS.

Candidates are required to use an operating system (PC (d)). The uses should relate to the range of features defined above, so the choice of operating system should enable the candidates to explore these features. It is unlikely that a simple OS (such as BBC DFS) would permit this. Not all of the defined features can be used (such as software compatibility) or will be supported in every OS (such as resource sharing) or can be explicitly applied (such as memory management, which is transparent in many operating systems). However, the candidate is expected to use a wide range of features and operating systems that do not permit candidates to demonstrate competence in many of the defined features are not suitable for this performance criterion. The use of each feature should be fairly sophisticated at this level. So trivial applications of any feature are not appropriate. For example, in the context of the user interface the candidate would be expected to use it efficiently and effectively in carrying out a wide range of tasks and also be able to customise the interface and automate common routines.

National Unit Specification: support notes (cont)

UNIT Computer Systems (Advanced Higher)

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

This is the last in a series of units relating to Computer Systems and, as such, candidates will already possess detailed knowledge of 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 ..... 10 hours
Outcome 2 ..... 15 hours
Outcome 3 ..... 15 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.

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 architecture of microprocessors.
Outcome 2	Extended-response questions on microprocessors and assembly language. Practical assignment involving the candidate in writing various assembly language program segments.
Outcome 3	Extended-response questions on the features of operating systems. Practical assignment on the use of a specific operating 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 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. For example, the written assessments (Outcomes 1, 2 and 3) could be combined into a single assessment activity covering all the knowledge and understanding contained within these outcomes. This would reduce the number of separate assessment activities to three - one written and two practical assessments.

Written evidence may take various forms including handwritten 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 Systems (Advanced 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 Data Communications (Advanced Higher)

NUMBER D103 13

COURSE Computing (Advanced Higher)

SUMMARY

This unit develops the ability of the candidate to analyse the hardware, software and systems associated with computer communication and use computer hardware and software to perform data communications.

OUTCOMES

1 Describe the characteristics of network application programs.
2 Describe the architecture of computer networks.
3 Use hardware and software to perform data communications.

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):

- Internetworking (H) unit
- Computing course at Higher level

CREDIT VALUE

1 credit at Advanced Higher.

Administrative Information

Superclass: CB

Publication date: December 1999

Source: Scottish Qualifications Authority

Version: 02

National Unit Specification: general information (cont)

UNIT Data Communications (Advanced Higher)

CORE SKILLS

Core skills for Advanced Higher remain subject to confirmation and details will be available at a later date.

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 Data Communications (Advanced Higher)

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

OUTCOME 1

Describe the characteristics of network application programs.

Performance criteria

(a) Development of network application programs is correctly described.
(b) Types of network application program are correctly described.
(c) Standards relevant to network application programs are correctly described.
(d) Descriptions are correctly related to specific network application programs.

Note on range for the outcome

Network application programs: information access; file transfer; e-mail; remote access.

Evidence requirements

Written or oral evidence that the candidate can describe the characteristics of network software as described in PCs (a) to (d) for all classes in the range.

OUTCOME 2

Describe the architecture of computer networks.

Performance criteria

(a) National and international standards relating to data communications are correctly described.
(b) Hardware components are correctly described.
(c) Protocols are accurately described.
(d) Descriptions are correctly related to a specific network system.

Note on range for the outcome

Computer network: local-area network; wide-area network; peer-to-peer; client-server.

Evidence requirements

Written or oral evidence that the candidate can describe the architecture of computer networks as detailed PCs (a) to (d) for all classes in the range.

National Unit Specification: statement of standards (cont)

UNIT Data Communications (Advanced Higher)

OUTCOME 3

Use hardware and software to perform data communications.

Performance criteria

(a) Hardware and software are configured correctly.
(b) Hardware and software are used efficiently and effectively.
(c) Operations are carried out efficiently and effectively.
(d) Network etiquette is known and observed.

Note on range for the outcome

Use: information access; file transfer; e-mail; remote access.

Evidence requirements

Performance evidence that the candidate can use hardware and software to perform data communications as detailed in PCs (a) to (d) for all classes in the range.

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

National Unit Specification: support notes

UNIT Data Communications (Advanced 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 Advanced Higher level. The details of content given earlier provide information on the delivery and assessment of this unit within a course context.

Outcome 1

This outcome relates to the characteristics of network programs. The candidate should be able to describe the historical development of network application programs (PC (a)). This will include a brief description of the development of the main types of network application program together with an explanation of the need for network application programs.

Candidates must describe the different types of network programs (PC (b)). For the purposes of this outcome, the main types of network program are:

- information access
- file transfer
- e-mail
- remote access

This software will vary from network to network and candidates need only provide a brief overview of each type of program.

Candidates must also describe contemporary standards for network application programs (PC (c)). These standards should relate to the types of program previously defined. For example, the standards applicable to e-mail include SMTP and POP3 and encoding schemes such as MIME and UUencoding. Note that generic communication standards (such as the ISO model) are described in the next outcome.

Candidates must relate these generic descriptions to an actual network. This may be a local-area network (such as Ethernet) or a wide-area network (such as the Internet). This description will relate to the network's historical development, the types of application software it supports and the standards it employs. For example, if the Internet is selected, then the candidate would describe its development, relate the generic program types to actual Internet software, and describe the current standards that relate to the Internet (such as TCP/IP, SMTP, HTTP).

National Unit Specification: support notes (cont)

UNIT Data Communications (Advanced Higher)

Outcome 2

This outcome relates to the architecture of computer networks. The architecture of four different network systems should be described. These are:

- local-area networks
- wide-area networks
- peer-to-peer networks
- client-server networks

Candidates are required to describe national and international standards that relate to data communications (PC (a)). The main international standard that directly or indirectly relates to all networks is the ISO Reference Model and candidates should be aware of the structure of this model and the general purpose of each layer. At the time of writing, the main protocols in current use are:

- TCP/IP
- FTP
- SMTP
- HTTP
- Z modem
- Kermit
- X.25
- V34

The candidate should be able to describe these protocols in the context of:

- encoding information
- flow control
- error control

Candidates must also be able to describe proprietary protocols that are used in many private networks (PC (c)). While most networks employ national or international standards for part of the network, proprietary systems (such as Ethernet and Token Ring systems) will also use alternative protocols for certain functions (such as Ethernet's CSMA/CD).

Candidates are required to relate these descriptions to a specific network (PC (d)). The candidate should be able to describe an existing network in terms of:

- topology
- standards (including protocols)
- hardware components
- bandwidth requirements
- communication software
- applications software
- cost

In particular, candidates must be able to relate the ISO Reference Model to this network by mapping each of the seven layers to that specific system's architecture.

Outcome 3

This outcome involves the candidate in using a network system. The specified uses are:

- information access
- file transfer
- e-mail mail
- remote access

A wide range of computer networks could be used. In the context of the Internet, information access would involve candidates in using browsing and newsreading software, file transfer would involve the use of an FTP client program, e-mail would involve the use of an Internet mail program and remote access would involve the use of Telnet (or equivalent) software.

The candidate should be able to configure hardware and software (PC (a)). In the context of a local-area network, this would involve setting various parameters relating to network performance using a range of network management software. In the context of the Internet, this would involve the configuration of a range of client software, such as browsers, readers, FTP clients and Telnet software.

Candidates must use hardware and software efficiently and effectively (PC (b)) and the operations must be carried out efficiently and effectively (PC (c)). This not only relates to the correct configuration of network hardware and software but also the competent use of the network's resources. So, for example, it would be unacceptable if candidates are unable to use a piece of software correctly, or use that software inefficiently.

Network etiquette is an informal code of conduct when using a computer network. Detailed network etiquette will be provided as part of the support materials. Candidates are required to observe 'netiquette' when using computer networks (PC (d)).

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 ..... 10 hours
Outcome 2 ..... 10 hours
Outcome 3 ..... 20 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 Data Communications (Advanced 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 characteristics of network application software.

Outcome 2 Extended-response questions on the architecture of computer networks.

Outcome 3 Practical assignment on the use of a computer network.
Extended-response questions on network etiquette.

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. For example, the written assessments (Outcomes 1, 2 and 3) could be combined into a single assessment activity covering all the knowledge and understanding contained within these outcomes. This would reduce the number of separate assessment activities to two - one written assessment and one practical assessment.

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).