National Unit Specification: general information

UNIT **Computer Architecture (Advanced Higher)**

NUMBER DM44 13

COURSE Computing (Advanced Higher)

SUMMARY

This Unit is designed to develop knowledge and understanding of the principles of computer architecture and provides an opportunity to apply this through the use of contemporary hardware and software. The overall theme of the Unit is the relationship between design and performance. This knowledge, understanding and related practical skills may then be applied by the candidate to solve practical problems related to computer systems.

It is designed as an option for candidates undertaking the Advanced Higher Computing Course, but is also suitable for anyone wishing to extend and deepen their experience of computer systems beyond Higher level.

OUTCOMES

1. Demonstrate knowledge and understanding of the main theories, concepts and principles relevant to the process of designing computer systems to maximise performance.
2. Demonstrate practical skills in the context of computer architecture using contemporary hardware and software.

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:

- *Computer Systems* (Higher)Unit
- Higher Computing

Administrative Information

Superclass: CA

Publication date: April 2005

Source: Scottish Qualifications Authority

Version: 1

© Scottish Qualifications Authority 2005

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

Additional copies of this unit specification can be purchased from the Scottish Qualifications Authority. The cost for each unit specification is £2.50. (A handling charge of £1.95 will apply to all orders for priced items.)

National Unit Specification: general information (cont)

UNIT Computer Architecture (Advanced Higher)

CREDIT VALUE

1 credit at Advanced Higher (8 SCQF points at SCQF level 7).*

*SCQF credit points are used to allocate credit to qualifications in the Scottish Credit and Qualifications Framework (SCQF). Each qualification in the Framework is allocated a number of SCQF credit points at an SCQF level. There are 12 SCQF levels, ranging from Access 1 to Doctorates.

CORE SKILLS

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

National Unit Specification: statement of standards

UNIT Computer Architecture (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 knowledge and understanding of the main theories, concepts and principles relevant to the process of designing computer systems to maximise performance.

Performance criteria

(a) A broad range of computing terminology is used appropriately.
(b) Descriptions and explanations are related to familiar and unfamiliar contexts.
(c) Conclusions, predictions and generalisations are made from knowledge and understanding.

Evidence requirements

Written or oral evidence that the candidate can describe and explain the principles and features relevant to the process of designing computer systems to maximise performance accurately and concisely. Evidence should be obtained using questions in a closed-book test, under supervision, lasting no more than 45 minutes. The test must sample across the range of the content (see Computing (Advanced Higher) Course content) in each of the following areas:

- computer structure
- processor structure
- processor development
- operating systems.

(The content statements are also reproduced for convenience as a table in the support notes for this Unit).

The standard to be applied is illustrated in the National Assessment Bank items available for this Unit.

If a centre wishes to design its own assessments for this Unit, they should be of a comparable standard.

National Unit Specification: statement of standards (cont)

UNIT Computer Architecture (Advanced Higher)

OUTCOME 2
Demonstrate practical skills in the context of computer systems using contemporary hardware and software.

Performance criteria

(a) A range of features of hardware is used effectively and efficiently.
(b) An appropriate range of features of software is used effectively and efficiently.
(c) Practical tasks are planned and organized independently.
(d) Practical tasks are undertaken in an appropriate range of non-routine contexts.

Evidence requirements

Observation checklist showing that the candidate has demonstrated the following practical skills in the context and at a level defined by the content statements (see Computing (Advanced Higher) Course content):

- comparing the facilities of two operating systems
- implementing simple assembly language instructions (using an assembler or an emulator).

A brief report on the comparison of two operating systems should be provided by the candidate.

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

The practical skills should be demonstrated in the context defined in the content statements (see Computing (Advanced Higher) Course content).

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

(The content statements are also reproduced for convenience as a table in the support notes for this Unit).

The standard to be applied is illustrated in the National Assessment Bank items available for this Unit.

If a centre wishes to design its own assessments for this Unit, they should be of a comparable standard.

National Unit Specification: support notes

UNIT Computer Architecture (Advanced Higher)

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

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

GUIDANCE ON THE CONTENT AND CONTEXT FOR THIS UNIT

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

Content Statements: Computer structure
Detailed description of processor and registers including MAR, MDR, IR, PC and general purpose registers. Description of relationship between buses and processor registers. Description of the fetch-execute cycle in terms of buses and registers.
Description of the following types of computer memory: ♦ internal memory (registers, cache , main memory) ♦ external memory (magnetic disk, CD-ROM, CD-RW, DVD-ROM, rewritable DVD, tape) Description of these memory technologies in a hierarchy, using the following factors: ♦ decreasing cost per bit ♦ increasing capacity ♦ increasing access time ♦ decreasing frequency of access Explanation of the importance of these factors when designing a system for performance.
Description of the structured use of cache memory to improve processor performance referring to the use of level 1 and level 2 cache as well as the use of static RAM.
Description of how memory interleaving operates. Description of how memory interleaving can improve system performance by enabling multiple memory accesses simultaneously.
Description of how direct memory access can improve system performance.
Description of the effect on system performance of increasing clock speeds and increasing the width of data buses with reference to 8 bit, 16 bit, 32 bit and 64 bit microprocessors.
Description of PCI and PCI-X buses in terms of the following characteristics: throughput described as bits per second, width, multipoint topology, function. Description of the importance of bus throughput for system performance.

Content Statements: Processor structure
Description and exemplification of the structure of assembly language instructions as op-code and operand.
Description and exemplification of the classification of assembly language instructions using the following categories: data transfer, arithmetic, logical, shift and rotate, branch.
Description of the following key features which distinguish RISC from CISC: a limited and simple instruction set, the use of register oriented instructions with limited memory access, the use of limited addressing modes and a large bank of registers.
Description of the performance gain derived from the use of SIMD (single instruction multiple data) instructions with reference to multimedia processing operations.

National Unit Specification: support notes (cont)

UNIT Computer Architecture (Advanced Higher)

Description of how the following techniques can be used to optimise the instruction and data stream: ♦ branch prediction ♦ data flow analysis ♦ speculative loading of data ♦ speculative execution of instructions ♦ predication
Description of how pipelining operates and how it can improve system performance. Description of possible problems with pipelining caused by branch instructions and by instructions of different lengths.
Descriptions of how superscalar processing operates and how it can improve system performance.

Content Statements: Processor development
Description of the evolution of the following microprocessor architectures: the Power PC series, the Intel X86 series and the Intel IA-64 in terms, where appropriate, of the following features and techniques: ♦ increasing clock speeds ♦ data bus widths ♦ pipelining ♦ superscalar processing ♦ branch prediction ♦ speculative loading of data and executing of instructions ♦ predication ♦ the number and function of registers used ♦ SIMD ♦ RISC ♦ CISC Explanation of the relationship between these developments and system performance.
Description of how parallel computers function referring to their use of: ♦ local (cache) as well as main memory ♦ pipelining ♦ local pathways and packet switching to achieve communication between CPUs. Description of the performance benefits of parallel computers.

Content Statements: Operating systems
Description of the following techniques used by operating systems to manage memory: ♦ variable partitioning of memory ♦ use of best-fit, worst fit and first fit algorithms for the allocation of memory. Comparison of the operation of best fit, worst fit and first fit algorithms in terms of efficient use of memory. Description of the relationship between the size of data blocks used in memory allocation and access speed.
Description of the need for operating systems to schedule programs in a multitasking system: Comparison of the following types of pre-emptive scheduling in terms of their effect on system performance: Round robin scheduling, multi-level feedback queue.

National Unit Specification: support notes (cont)

UNIT Computer Architecture (Advanced Higher)

Description of the use of direct memory access to manage input/output data transfers in order to improve system performance.
Description of the key function of the file management system as mapping between the logical view of files and their physical location. Description of contiguous and non-contiguous methods of allocation of files to available storage space.
Description of the way in which the trend towards designing GUI based on the design principle of 'convenience for the user' leads to: ♦ increasing software complexity ♦ more demands on system resources (processor and memory) ♦ a consequent burden on system performance Illustration of the demands a GUI makes on the system resources by describing the steps involved in processing a simple operation such as a mouse click. Comparison with the demands a CLI makes on the system.
Description of the trend to expand the role of operating systems to include services and provide capabilities that were formerly within applications themselves. Description of those services as: ♦ providing a standard look and feel for applications ♦ simplifying and extending graphic capabilities of application programs ♦ improving the capability of programs to communicate and pass data ♦ the capability of launching one application inside another. Description of the use of libraries of objects that applications call as required. Description of techniques used by OS to support communication between applications and document embedding.
Description of procedures used to control user access in both multi-user and single user systems. Description of the purpose of file attributes. Description of backup facilities.
Exemplification and comparison of the features of **two** specific operating systems.

GUIDANCE ON LEARNING AND TEACHING APPROACHES FOR THIS UNIT

Candidates will require individual access to appropriate computer hardware and software throughout this Unit.

The two Outcomes should be delivered in an integrated way rather than sequentially. For Outcome 2, the practical activities should be taught and used to illustrate and exemplify the knowledge and understanding required for Outcome 1. The underlying theme of 'design for performance' should be illustrated and exemplified throughout the Unit.

Although most of the work of this Unit can be completed using a single computer system, candidates will require some access to an alternative operating system. They will also require a level of access which allows making changes to operating system parameters. Practical work involving assembly language could be undertaken using a real assembler or using a simulation program.

National Unit Specification: support notes (cont)

UNIT Computer Architecture (Advanced Higher)

The amount of time spent on each area of content will vary depending on the teaching methodology used and the ability and prior experience of the candidates. However, the following times may be used as a rough guide:

computer structure	8 hours
processor structure	8 hours
processor development	6 hours
operating systems	14 hours

1½ hours should be set aside to:

- administer the Outcome 1 test
- gather evidence for Outcome 2.

A further 2½ hours is allowed for remediation and re-assessment if required.

If the Unit is delivered as part of a Course, the Course documentation will provide further information on teaching and learning in a Course context, including the identification of a number of 'themes' to facilitate holistic learning across the Course.

GUIDANCE ON APPROACHES TO ASSESSMENT FOR THIS UNIT

National Assessment Bank tests have been created specifically to assess Outcome 1 of the Unit. This assessment consists of a closed-book test, and must be conducted under supervision. In order to gain success in this Outcome, the candidate must achieve at least the cut-off score for the test. If a centre wishes to design its own assessments for this Unit, they should be of a comparable standard.

Outcome 2 requires the candidate to demonstrate practical skills while using contemporary hardware and software. These practical skills may be demonstrated in a single extended task or a number of relatively small tasks. The task(s) will normally be undertaken by the candidate as part of the teaching and learning activities of the Unit, rather than as separate formal assessment activities. The candidate will be allowed access to books, notes and on-line help while completing the task(s). The practical skills should be demonstrated in the context defined in the content statements (see Computing (Advanced Higher) Course content).

To gain success in this Outcome, the candidate must demonstrate practical skills at an appropriate level in both of the following contexts:

- comparing the facilities of two operating systems
- implementing simple assembly language instructions (using an assembler or an emulator).

A brief report on the comparison of two operating systems should be provided by the candidate.

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

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

National Unit Specification: support notes (cont)

UNIT Computer Architecture (Advanced Higher)

CANDIDATES WITH ADDITIONAL SUPPORT NEEDS

This Unit Specification is intended to ensure that there are no artificial barriers to learning or assessment. The additional support needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments or considering alternative Outcomes for Units. For information on these, please refer to the document *Guidance on Assessment Arrangements for Candidates with Disabilities and/or Additional Support Needs* (SQA, 2004).