## Higher National Unit specification: general information

| | |
|---|---|
| **Unit title:** | Software Development: Systems Foundation |
| **Unit code:** | H17Y 34 |

| | |
|---|---|
| **Superclass:** | CB |
| **Publication date:** | March 2012 |
| **Source:** | Scottish Qualifications Authority |
| **Version:** | 01 |

## Unit purpose

This Unit is designed to develop further understanding and use of operating systems, and to introduce candidates to utility software, task automation and memory management. The Unit should build on the knowledge of operating systems and utility software learned in the HNC Computing core Unit H175 34 *Computer Systems Fundamentals,* introducing the new concepts of low level language programming and shell scripting.

On completion of the Unit the candidate should be able to:

1 Develop a strategy to use utility software and explain how it functions in the context of an operating system.
2 Construct a small program which uses instructions and memory registers in a low-level language.
3 Develop shell scripts to automate routine tasks.

## Recommended prior knowledge and skills

Access to this Unit will be at the discretion of the Centre. However it would be beneficial if the candidate already possessed good written communication, critical thinking and analytical skills, either through workplace experience or training at an appropriate level. It would also be beneficial if candidates had some prior experience of operating systems and utility software which could be evidenced by the HN Unit H175 34 *Computer Systems Fundamentals,* and some prior experience of the basic programming constructs which could be evidenced by the HN Unit H173 34 *Developing Software: Introduction*

## General information (cont)

## Credit points and level

2 Higher National Unit credits at SCQF level 7: (16 SCQF credit 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

Opportunities to develop aspects of Core Skills are highlighted in the Support Notes of this Unit specification.

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

## Context for delivery

If this Unit is delivered as part of a Group Award, it is recommended that it should be taught and assessed within the subject area of the Group Award to which it contributes. It is recommended that this Unit is delivered either in conjunction with, or after completion of, the HNC Computing core Units H175 34 *Computer Systems Fundamentals* and H173 34 *Developing Software: Introduction*.

## Higher National Unit specification: statement of standards

## Unit title: Software Development: Systems Foundation

## Unit code: H17Y 34

The sections of the Unit stating the Outcomes, Knowledge and/or Skills, and Evidence Requirements are mandatory.

Where evidence for Outcomes is assessed on a sample basis, the whole of the content listed in the Knowledge and/or Skills section must be taught and available for assessment. Candidates should not know in advance the items on which they will be assessed and different items should be sampled on each assessment occasion.

## Outcome 1

Develop a strategy to use utility software and explain how it functions in the context of an operating system.

### Knowledge and/or Skills

- Operating systems
- Utility software
- File management
- Memory management

### Evidence Requirements

Candidates will be required to demonstrate their Knowledge and Skills by developing a strategy to maintain data security and integrity, and use utilities to carry out that strategy. The strategy should be documented. The carrying out of the strategy may be documented or alternatively may be observed.

The assessment for this Outcome should be conducted under supervised open-book conditions.

## Unit title: Software Development: Systems Foundation

## Outcome 2

Construct a small program which uses instructions and memory registers in a low-level language.

### Knowledge and/or Skills

- flow control
- use of the accumulator
- managing registers

### Evidence Requirements

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can create a program of around 30–50 lines which makes use of the accumulator, manages registers, and uses flow control. This could take the form of a completion exercise.

The assessment for this Outcome should be conducted under supervised open-book conditions.

## Outcome 3

Develop shell scripts to automate routine tasks.

### Knowledge and/or Skills

- Sequence
- Iteration
- Selection
- Input and output
- Invocation of external applications or utilities

### Evidence Requirements

Candidates will need to provide evidence to demonstrate their Knowledge and/or Skills by showing that they can create a simple shell script which has input, output, and invocation of an external application or utility software.

Suitable examples would include the moving of files with a particular extension to a designated folder, or the mapping of network drives.

This assessment for this Outcome should be conducted under supervised open-book conditions.

## Higher National Unit specification: support notes

## **Unit title:** Software Development: Systems Foundation

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 80 hours.

## Guidance on the content and context for this Unit

This Unit has been designed to complement the core HNC Computing Unit H175 34 *Computer Systems Fundamentals*. It aims to introduce candidates to low level programming and shell scripting for the automation of common operating system and network management tasks.

Outcome 1 complements H175 34 *Computer Systems Fundamentals,* in looking at the use of utility software in the context of an operating system. The focus is on important utility software such as anti-virus software, file compression, and disk defragmentation.

Outcome 2 introduces candidates to low level programming, and its role in memory management. It is recommended that Assembler is used for this, but has been written flexibly so that any centre wishing to use machine code for this Outcome may do so.

Outcome 3 introduces candidates to simple shell scripting and how these can be used to automate simple systems tasks such as drive mapping.

Although it is possible to deliver this Unit as a standalone Unit, it may prove beneficial to combine it with a program development Unit such as H173 34 *Developing Software: Introduction*. Candidates will benefit from the prior exposure to basic programming constructs which will overlap with the skills required for the shell scripting in Outcome 3, and would also be beneficial for Outcome 2. It would also be beneficial to deliver this in conjunction with, or following, H175 34 *Computer Systems Fundamentals* as there is some useful overlap with the skills required for Outcome 1.

This Unit covers some of the skills described for a pre entry/Junior technician role in the National Occupational Standards — IT and Telecoms (2009). The main areas covered correspond to discipline 6.2 IT Security Management and discipline 7.5 IT/Technology Management and Support. There are also ample opportunities within the Unit to address a range of skills at both foundation and intermediate level that are described in the National Occupational Standards for IT Users v3. The most likely areas to be covered would be Optimise System Performance and IT Security for Users.

## Unit title: Software Development: Systems Foundation

## Guidance on the delivery of this Unit

This Unit has been designed to allow centres flexibility over which operating system, which utility software programs, and which low level and scripting languages are used for delivery. For example, Outcome 1 could be done using a Windows, Mac or Linux-based operating system, and appropriate compatible utility software. Outcome 2 is likely to involve the use of Assembler, but the version of Assembler is at the discretion of the centre. By referring to 'low level programming' rather than specifying Assembler, it also leaves open the possibility of centres choosing to deliver Outcome 2 using machine code. Outcome 3 could be delivered using a number of different shell scripting languages, depending on the operating system used.

### Outcome 1

Outcome 1 is a mixture of theory and practice. It is expected that this would cover the utilities used to maintain an operating system's performance. You could concentrate on important operating system tasks such as file management: how are files arranged on disks, what different file protocols can be used (FAT32, NTFS and others) and the advantages and disadvantages of each, why fragmentation occurs and how file defragmentation works, and the risks posed by malware and the strategies used by tools to counter them. This then leads on to file compression. It would be good to show candidates an example of a very simple compression algorithm (eg searching for the word 'the' in a file and replacing each instance of 'the' with the number 1). Candidates could then be given the practical task of implementing their own simple compression algorithm. This should be possible either after they have completed Outcome 2, or on the basis of what they have learned in the core HNC Computing Unit H173 34 *Developing Software: Introduction*. Disk defragmentation can then lead on to memory management and memory fragmentation. Memory management (how memory is structured and accessed) and pointers could be covered at this stage. The level of detail would depend on how much of this had already been covered in H175 34 *Computer Systems Fundamentals.* Finally, malware could be covered in this Outcome. Whilst it would be inadvisable to teach candidates how to create computer malware, it would be helpful for them to understand the basics of how malware programs are structured, and how software can detect and destroy them.

### Outcome 2

Outcome 2 is designed to be a practical introduction to low level programming. This could tie in with the discussion of memory management in Outcome 1. It is recommended that some flavour of Assembler is used. Candidates should be introduced to the basic instruction set: MOV, ADD, SUB, MUL, DIV, and the use of Labels, JMP, JZ and JNZ. Use of the accumulator, management of registers, and flow control would be covered. If using an assembler environment that permits visualisation (such as http://www.softwareforeducation.com/), then a suitable example programs could be based on the management of traffic lights, which shows flow control and the management of multiple registers. The Fetch-Execute cycle could also be introduced as a basis for this practical Outcome.

## Unit title: Software Development: Systems Foundation

### Outcome 3

Outcome 3 is designed to be a practical introduction to basic shell scripting. It is expected that candidates will already have some knowledge of programming basics through the core HNC Computing Unit H173 34 *Developing Software: Introduction*. Candidates should be introduced to input, output, and invocation of external applications or utilities through shell scripts. Highlight how shell scripts can be used to automate tasks such as mapping drives, mapping printers, moving files, compressing files over a certain size, and so on. As the internal command sets for operating systems are very small (eg see http://www.computerhope.com/jargon/i/intecomm.htm for the MS DOS command set), use of external commands (see http://www.computerhope.com/jargon/e/extecomm.htmfor the MS DOS external commands) will be required for the majority of scripts. Examples of suitable scripts could include log rotation and compression of files over a certain size with a certain extension type.

While the exact time allocated to this Unit is at the discretion of the centre, the notional design length is 80 hours. A suggested allocation of time is:

| | |
|---|---|
| Operating Systems | 4 hours |
| Viruses and anti-virus software | 6 hours |
| File management systems and fragmentation | 6 hours |
| Memory management and pointers | 12 hours |
| Assembler | 22 hours |
| Shell scripting | 22 hours |
| Assessment | 8 hours |

## Guidance on the assessment of this Unit

All assessments are practical in nature, and could be undertaken using e-assessment. Outcome 1 requires candidates to carry out a physical task, but the documentation for this could be submitted as an e-portfolio. Outcomes 2 and 3 involve the creation of small computer programs or scripts, which can also be submitted as part of an e-portfolio.

**Unit title:** Software Development: Systems Foundation

## Assessment Guidelines

All assessments are open-book but must be undertaken in supervised conditions.

### Outcome 1

Evidence for understanding fundamental aspects of operating systems can be produced by candidates carrying out, explaining and documenting a suitable task. For example, they could create a guide to defragmenting a hard disk drive, complete with step-by-step instructions, screenshots, and explanations.

### Outcome 2

Evidence for understanding the basics of programming in a low level language could be achieved by one working program, or through a portfolio of two or more programs that together cover the Evidence Requirements.

### Outcome 3

Evidence for understanding the basics of shell scripting could be achieved by one working program, or through a portfolio of two or more programs that together cover the Evidence Requirements. Ideally, the script would automate a process such as moving files or mapping drives. This Outcome could be integrated with the Unit H173 34 *Developing Software: Introduction.* It is also feasible that the script could be used to automate the task set for Outcome 1, therefore enabling the assessments for Outcome 1 and Outcome 3 to be combined.

## Online and Distance Learning

This Unit may be suitable for delivery for distance or online learning. The practical aspects of the Unit could be demonstrated and assessed by the utilisation of screen capture software, and the programming and scripting can be carried out using freely available IDEs and submitted electronically.

## Opportunities for the use of e-assessment

E-assessment may be appropriate for some assessments in this Unit. By e-assessment we mean assessment which is supported by Information and Communication Technology (ICT), such as e-testing or the use of e-portfolios or social software. Centres which wish to use e-assessment must ensure that the national standard is applied to all candidate evidence and that conditions of assessment as specified in the Evidence Requirements are met, regardless of the mode of gathering evidence. Further advice is available in *SQA Guidelines on Online Assessment for Further Education (AA1641, March 2003).*

**Unit title:** Software Development: Systems Foundation

## Opportunities for developing Core Skills

There may be opportunities to gather evidence towards Core Skills in this Unit, although there is no automatic certification of Core Skills or Core Skills components.

Candidates have the opportunity to develop the Core Skill component of critical thinking through identifying how to design working shell scripts and low level programs based on a problem statement which does not provide detailed instructions on how to design or structure a solution.

## Disabled candidates and/or those with additional support needs

The additional support needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments, or considering whether any reasonable adjustments may be required. Further advice can be found on our website **www.sqa.org.uk/assessmentarrangements**

## History of changes to Unit

| Version | Description of change | Date |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Additional copies of this Unit specification can be purchased from the Scottish Qualifications Authority. Please contact the Business Development and Customer Support team, telephone 0303 333 0330.

## General information for candidates

## **Unit title:** Software Development: Systems Foundation

This Unit is designed to enable you to develop a knowledge of utility software, file and memory management, in the context of an operating system. The Unit develops your practical systems administration skills through tasks such as disk defragmentation.

The Unit will also introduce you to low level language programming and how this is involved with memory management. The Unit further develops your practical systems administration skills through an introduction to shell scripting and the automation of simple systems tasks such as drive mapping using scripts.

The study of this Unit complements the core HNC Computing Units H175 34 *Computer System Fundamentals* and H173 34 *Developing Software: Introduction*, and provides a strong foundation for anyone interested in systems administration.

On completion of this Unit you should be able to:

1 Develop a strategy to use utility software and explain how it functions in the context of an operating system.
2 Construct a small program which uses instructions and memory registers in a low-level language.
3 Develop shell scripts to automate routine tasks.