

Higher National Unit Specification

General information for centres

Unit title: Scripting

Unit code: DG05 34

Unit purpose: This Unit is designed to introduce candidates to Scripting. It is intended for candidates undertaking an HNC or HND in Computing, Computer Networking or a related area who require an understanding of GUI and command-line scripting languages. On completion of the Unit candidates should be able to:

1. Describe scripting environment.
2. Describe GUI scripting language concepts.
3. Describe command-line scripting language concepts.
4. Write simple scripts.
5. Handle I/O and files.
6. Write complex scripts.

Credit value: 2 HN credit 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.*

Recommended prior knowledge and skills: Access to this Unit will be at the discretion of the Centre. There are no specific requirements but candidates would benefit from a basic knowledge of PC hardware and software. This may be demonstrated by the possession of HN Units such as DG0K 33 Hardware Concepts and DF9L 33 Operating System Concepts.

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.

Context for delivery: This Unit is included in the framework of a number of HNC and HND group awards. It is recommended that it should be taught and assessed within the context of the particular group award to which it contributes.

Assessment: Evidence for the knowledge and/or skills for the entire Unit must be produced using a set of 50 restricted-response questions to assess candidates' knowledge and understanding. This may be administered as a single end-of unit test, or as several subtests, each covering one or more outcomes.

Candidates must answer at least 70% of the questions correctly in order to obtain a pass. If subtests are used, they must also score at least 70% in each subtest.

Higher National Unit Specification: general information for centres (cont)

Testing must take place in a closed-book environment where candidates have no access to books, handouts, notes or other learning material. Testing can be done in either a machine-based or paper-based format and must be invigilated by a tutor or mentor. There must be no communication between candidates and communication with the invigilator must be restricted to matters relating to the administration of the test.

If a candidate requires to be reassessed, a different selection of questions must be used. At least half the questions in the reassessment must be different from those used in the original test.

If an outcome has a practical component, this must be assessed by having the candidate produce short scripts, illustrating particular scripting features. The scripts may be submitted in paper or electronic form and must be authenticated by the tutor or mentor.

For some outcomes only a sample of the practical tasks needs to be completed and recorded for assessment purposes, e.g. three out of five. This is clearly indicated in the instructions for the outcomes involved. Where this occurs, tutors must inform candidates of the tasks to be completed.

Higher National Unit specification: statement of standards

Unit title: Scripting

Unit code: DG05 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

Describe the scripting environment.

Knowledge and/or skills

- ◆ Operating system script host components.
- ◆ Operating system script host architecture.
- ◆ Using and running scripts.

Evidence requirements

The knowledge and skills component of Outcome 1 must be examined by nine questions, three being derived from each of the three items listed below. Each question must be derived from a single item.

1. Operating system script host components.

GUI environment, command-line environment, scripting controls, scripting engines.

2. Operating system script host architecture

Core object model, scripting hosts, scripting engines.

3. Using and running scripts.

Creating scripts, GUI execution, command-line execution.

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 9 questions for this outcome may contribute towards a single end-of-unit test of 50 questions.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

Short scripts

The restricted-response questions should be supplemented by a number of short scripts, illustrating particular Scripting features. In the case of this outcome, no short scripts are required.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 110 minutes should be allocated for a 50-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 2

Describe GUI scripting language concepts.

Knowledge and/or skills

- ◆ Data types.
- ◆ Variables and constants.
- ◆ Operators.
- ◆ Program flow.
- ◆ Procedures.

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 2 must be examined by ten questions, two being derived from each of the five items listed below. Each question must be derived from a single item.

1. Data types

Variants, subtypes, conversion functions, VarType.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

2. Variables and constants

Declaration, naming rules, scalar variables, array variables, scope, lifetime, multi-dimensional arrays, dynamic arrays, intrinsic constants, user-defined constants.

3. Operators

Arithmetic, comparison, concatenation, logical operators, operator precedence, parentheses.

4. Program flow

Conditional statements (If ... Then ... Else, Select Case), loops (Do ... Loop, While ... Wend, For ... Next, For Each ... Next)

5. Procedures

Sub procedures, Function procedures, returning values.

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 10 questions for this outcome may contribute towards a single end-of-unit test of 50 questions.

Short scripts

The restricted-response questions must be supplemented by a number of short scripts, illustrating particular Scripting features. Scripts must be functional, but may be trivial. In the case of this outcome, one or more short scripts must be produced to illustrate each of the following language features: data types, variables and constants, operators, flow control, procedures.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 110 minutes should be allocated for a 50-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

Outcome 3

Describe command-line scripting concepts.

Knowledge and/or skills

- ◆ Writing command-line scripts.
- ◆ Variables.
- ◆ Data types.
- ◆ Operators.
- ◆ Program flow.
- ◆ Functions.
- ◆ Objects.
- ◆ Reserved words.

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 3 must be examined by eight questions, derived from the eight items listed below. Each question must be derived from a different item.

1. Writing command-line scripts.

Statements, comments, assignments, equality, expressions.

2. Variables

Declaring variables, constructing variable names, null variables, undefined variables, implicit declaration, coercion.

3. Data types

Numbers, strings, objects, Booleans, null, undefined.

4. Operators

Arithmetic, logical, bitwise, assignment, miscellaneous, operator precedence, parentheses.

5. Program flow

Conditional statements (if, if ... else, implicit), loops (for, for ... in, while, do ... while, switch), break, continue

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

6. Functions

Built-in functions, user-defined functions, parameter passing.

7. Objects

Properties, methods, intrinsic objects, user-defined objects, browser objects, objects and arrays, subscript numbering

8. Reserved words

Protected reserved words, new reserved words, future reserved words

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 8 questions for this outcome may contribute towards a single end-of-unit test of 50 questions.

Short scripts

The restricted-response questions should be supplemented by a number of short scripts, illustrating particular scripting features. Scripts should be functional, but may be trivial. In the case of this outcome, one or more short scripts should be produced to illustrate each of the following language features: data types, variables, operators, program flow, functions, objects.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 110 minutes should be allocated for a 50-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 4

Write simple scripts.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

Knowledge and/or skills

- ◆ Creating scripts.
- ◆ Running scripts.
- ◆ Creating batch scripts.
- ◆ Displaying text strings.
- ◆ Examining script information.
- ◆ Working with environment variables.
- ◆ Running programs from within scripts.
- ◆ Combining scripts written in different languages.

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 4 must be examined by eight questions, one being derived from each of the eight items listed below. Each question must be derived from a different item.

1. Creating scripts

Using a text editor, file extensions

2. Running scripts

Scripting hosts (GUI, command-line)

3. Creating batch scripts

batch file extension, XML (Extensible Markup Language)

4. Displaying text strings

Display commands

5. Examining script information

Script host version, script engine version, location of script hosts, arguments passed, environment variables.

6. Working with environment variables

Types of environment variable

7. Running programs from within scripts

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

Applications, command-line programs, command shell scripts

8. Combining scripts written in different languages

Batch scripts, calling functions, returning values

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 8 questions for this outcome may contribute towards a single end-of-unit test of 50 questions.

Short scripts

The restricted-response questions must be supplemented by a number of short scripts, illustrating particular Scripting features. Scripts must be functional, but may be trivial. In the case of this outcome, one or more short scripts must be produced to illustrate at least three of the following features: displaying text strings, examining script information, working with environment variables, running programs from within scripts, combining different languages. The scripts to be written must be selected by the tutor. Scripts may be written in any scripting language or a combination of languages.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 110 minutes should be allocated for a 50-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 5

Handle I/O and files.

Knowledge and/or skills

- ◆ Input and Output concepts.
- ◆ Using Input Boxes and Message Boxes.
- ◆ Error detection and handling.
- ◆ File system objects.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

- ◆ Working with folders.
- ◆ Working with files.
- ◆ Reading text files.
- ◆ Writing to a file.

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 5 must be examined by eight questions, one derived from each of the items listed below. Each question must be derived from a single item.

1. Input and Output concepts

Standard I/O streams, StdIn, StdOut, StdErr

2. Using Input Boxes and Message Boxes

Input Boxes, Message Boxes, Buttons, Icons

3. Error detection and handling

Runtime errors, checking for and catching errors, examining error codes and descriptions

4. File system objects

Objects, collections

5. Working with folders

Viewing folder contents, examining and working with folder properties, checking for and creating folders, deleting, copying, and moving folders, working with special folders

6. Working with files

Examining and working with file properties, copying, moving, and deleting files, checking for and creating files, reading and writing files

7. Reading text files

Opening for reading, writing or appending, read entire contents, read character strings, read lines of information, skip characters or lines.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

8. Writing to a File

Writing, appending

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 8 questions for this outcome may contribute towards a single end-of-unit test of 50 questions.

Short scripts

The restricted-response questions must be supplemented by a number of short scripts, illustrating particular Scripting features. Scripts must be functional, but may be trivial. In the case of this outcome, one or more short scripts must be produced to illustrate at least five of the following features: using input boxes and message boxes, error detection and handling, file system objects, working with folders, working with files, reading text files, writing to a file. The scripts to be written must be selected by the tutor. Scripts may be written in any scripting language or a combination of languages.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 110 minutes should be allocated for a 50-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Outcome 6

Write complex scripts.

Knowledge and/or skills

- ◆ Managing drives.
- ◆ Managing network printers.
- ◆ Managing shortcuts and menus.
- ◆ Adding and removing startup applications.
- ◆ Working with the Windows registry.
- ◆ Writing to event logs.
- ◆ Reading event logs.

Evidence requirements

Restricted response questions

The knowledge and skills component of Outcome 6 must be examined by seven questions, derived from the seven items listed below. Each question must be derived from a different item.

1. Managing drives

Specific drives, drive collections, drive letter, drive path, network share path, drive designator, getting drive name, ensuring that drive exists, drive properties, getting drive designator, ascertaining free space, mapping network drive, removing network drive.

2. Managing network printers,

Adding and removing network printers, setting default printer, adding a printer connection, removing a printer connection.

3. Managing shortcuts and menus

Adding shortcuts to a user's programs or startup folder, using scripts to create, update and delete menus.

4. Adding and removing startup applications

Creating shortcuts, startup folders for single user and all users

5. Working with the system data

Read, write, and delete system data entries, reconfigure network services (DHCP, WINS)

6. Writing to event logs

Event logs (application, security, system, directory service, DNS server, file replication service), monitor critical systems, create customised reports.

7. Reading event logs

Log-searching and extraction routines, event log utilities

The test may be administered on its own as a subtest or be combined with other outcome subtests in the Unit.

Alternatively, the 7 questions for this outcome may contribute towards a single end-of-unit test of 50 questions.

Higher National Unit specification: statement of standards (cont)

Unit title: Scripting

Short scripts

The restricted-response questions must be supplemented by a number of short scripts, illustrating particular scripting features. Scripts must be functional, but may be trivial. In the case of this outcome, one or more short scripts must be produced to illustrate at least four of the following features: managing drives, managing network printers, managing shortcuts and menus, adding and removing startup applications, working with the windows registry, writing to event logs, reading event logs. The scripts to be produced must be selected by the tutor. Scripts may be written in any scripting language or a combination of languages.

Assessment guidelines

It is suggested that all the above concepts be presented and explained within the context of current real-world practice and applications.

The suggested time allocation for a restricted response test is 2 minutes for each question plus 5 minutes starting-up time and 5 minutes finishing-off time, thus a total of 110 minutes should be allocated for a 50-question end-of-unit test.

Although individual outcome tests are permissible, it is suggested that if subtests are to be used, outcomes should be combined to produce tests of no fewer than 10 questions. A 10-question test would therefore have a time allocation of 30 minutes.

Administrative Information

Unit code:	DG05 34
Unit title:	Scripting
Superclass category:	CD
Date of publication:	May 2004
Version	01
Source:	SQA

© Scottish Qualifications Authority 2004

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

Additional copies of this Unit specification can be purchased from the Scottish Qualifications Authority. The cost for each Unit specification is £2.50 (minimum order £5.00). Additional copies of the Assessment Exemplar, and Guidelines on the Delivery of the Unit can also be purchased from Scottish Qualifications Authority. The cost is £15.00.

Higher National Unit specification: support notes

Unit title: Scripting

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.

The suggested time allocation for each outcome (including assessment) is as follows:

Outcome 1:	5 hours
Outcome 2:	10 hours
Outcome 3:	10 hours
Outcome 4:	20 hours
Outcome 5:	15 hours
Outcome 6:	20 hours

Guidance on the content and context for this Unit

As it is likely that the bulk of the material in this Unit will be delivered through lecturer exposition, it is important that every opportunity is taken to introduce real-world examples, opportunities for whole-class and group discussion and practical demonstrations wherever possible. Concepts and terminology should be presented in context throughout the Unit. Video presentations should be used where appropriate for providing an alternative explanation of a difficult topic, or as a focus for class discussion or groupwork.

Given the theoretical nature of this Unit, it is intended that a significant amount of time will be made available as a central part of the course for revision, tutorials and formative assessment exercises. Candidates should be strongly encouraged to undertake further reading, and opportunities for individual or group research should be provided.

The most important overall emphasis should be on the relevance and currency of content in such a rapidly-evolving field.

These Support Notes provide examples based on the use of the VBScript and Jscript languages within a Windows environment, but no restriction is placed on the scripting languages to be used or the environment in which they are run. However, centres should note that changes to languages or environment may involve significant changes in terminology.

Outcome 1

1 Operating system script host components.

Candidates should be aware that Windows Script Host is a core component of the Windows 2000 operating system and is installed by default. They should know how to use **cscript** at a command prompt to obtain version information and usage details and should know that the key components of Windows Script host are:

Wscript: a Windows executable for the scripting host, used to execute scripts from the desktop and display output in pop-up dialog boxes.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Cscript: a command-line executable for the scripting host used to execute scripts from the command line and display standard output at the command line.

WSH ActiveX Control: an ActiveX control that provides the core object model for the scripting host.

Scripting Engines: provide the core functions, objects, and methods for a particular scripting language. VBScript and JScript are installed by default on Windows 2000.

2. Operating system script host architecture.

Candidates should be aware that the WSH.ocx ActiveX control allows scripts to interact with the operating system. In WSH, objects are named containers, used to interact with OS components, eg: the WshNetwork object is used to access and configure network resources. They should be aware that each object has properties and methods that can be used to perform tasks. Properties are attributes of an object that can be accessed, methods are procedures used to perform operations. They should be familiar with exposed objects, which are used in scripts and non-exposed objects, which work in the background.

Candidates should know that there are two scripting hosts available, WScript and CScript. WScript has GUI controls for displaying output in pop-up dialog boxes and is used to execute scripts from the desktop. CScript is used to execute scripts from the command line. They should know that batch scripts contain markup tags that identify elements within the batch, such as individual jobs and the scripting language being used. Markup tags are defined as XML (Extensible Markup Language) elements

Candidates should know that scripting engines provide the core language functionality for Windows scripts. Scripting engines are available for JScript, VBScript, Perl, TCL, Python, and others. VBScript and JScript are standard components from Windows 2000 onwards.

Outcome 2

The principal GUI-based scripting language used in a Windows environment is VBScript.

1. Data types

Candidates should know that VBScript has only one data type called a **Variant**. A **Variant** is a special kind of data type that can contain different kinds of information, depending on how it's used. **Variant** is also the data type returned by all functions in VBScript. At its simplest, a **Variant** can contain either numeric or string information. A **Variant** behaves as a number when used in a numeric context and as a string when used in a string context.

A **Variant** can make further distinctions about the specific nature of numeric information, e.g. numeric information can represent a date or a time. When used with other date or time data, the result is always expressed as a date or a time. The different categories of information that can be contained in a **Variant** are called subtypes.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Conversion functions can be used to convert data from one subtype to another. In addition, the **VarType** function returns information about how data is stored within a **Variant**.

2. Variables and constants

Candidates should be aware that variables should be declared explicitly using **Dim**, **Public** or **Private** statements. Implicit declarations can be avoided by using the **Option Explicit** statement.

Variable names follow the standard rules for naming anything in VBScript. They must begin with an alphabetic character, cannot contain an embedded period, must not exceed 255 characters and must be unique in the scope in which they are declared.

A variable's scope is determined by where it is declared. If a variable is declared within a procedure, only code within that procedure can access it or change its value. It has local scope and is called a procedure-level variable. If a variable is declared outside a procedure, it is a script-level variable, and has script-level scope.

The lifetime of a variable is the time period during which it exists. The lifetime of a script-level variable extends from the time it is declared until the time the script is finished running. At procedure level, a variable exists only as long as the procedure is executing. When the procedure exits, the variable is destroyed.

A variable containing a single value is a scalar variable. It is also possible to create variables that can contain a series of values. This is called an array variable. Array variables and scalar variables are declared in the same way, except that the declaration of an array variable uses parentheses () following the variable name. Arrays can have up to 60 dimensions. An array whose size changes during the time a script is running is called a dynamic array.

A constant is a meaningful name that takes the place of a number or string and never changes. VBScript defines a number of intrinsic constants. The **Const** statement can be used to create string or numeric constants and assign them literal values.

3. Operators

Candidates should be aware that VBScript has a full range of operators, including arithmetic operators, comparison operators, concatenation operators and logical operators. When several operations occur in an expression, each part is evaluated and resolved in a predetermined order called operator precedence. Parentheses can be used to override the order of precedence and force some parts of an expression to be evaluated before others. Operations within parentheses are always performed before those outside. Within the parentheses standard operator precedence is maintained.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

4. Flow control

Candidates should be aware that the flow of a script can be controlled by using conditional statements and looping statements. Conditional statements can be used to write VBScript code that makes decisions and repeats actions. The conditional statements available in VBScript are **If ... Then ... Else** statement and the **Select Case** statement.

The **If ...Then...Else** statement is used to evaluate whether a condition is **True** or **False** and, depending on the result, to specify one or more statements to run. **If ... Then ... Else** statements can be nested to as many levels as required.

Select Case works with a single test expression that is evaluated once, at the top of the structure. The result of the expression is compared with the values for each **Case** in the structure. If there is a match, the block of statements associated with that **Case** is executed.

Looping allows a group of statements to be run repeatedly. Some loops repeat statements until a condition is **False**; others repeat statements until a condition is **True**. There are also loops that repeat statements a specific number of times. The following looping statements are available in VBScript:

- **Do ... Loop**: Loops while or until a condition is **True**.
- **While ... Wend**: Loops while a condition is **True**.
- **For ... Next**: Uses a counter to run statements a specified number of times.
- **For Each ... Next**: Repeats a group of statements for each item in a collection or each element of an array.

5. Procedures

Candidates should be aware that there are two kinds of procedures in VBScript, the **Sub** procedure and the **Function** procedure.

A **Sub** procedure is a series of VBScript statements, enclosed by **Sub** and **End Sub** statements that perform actions but don't return a value. A **Sub** procedure can take arguments (constants, variables or expressions) passed by a calling procedure.

A **Function** procedure is a series of VBScript statements enclosed by the **Function** and **End Function** statements. A **Function** procedure is similar to a **Sub** procedure, but can also return a value. A **Function** procedure can also take arguments passed to it by a calling procedure. A **Function** returns a value by assigning a value to its name in one or more statements of the procedure. The return type of a **Function** is always a **VARIANT**. The only way to get data out of a procedure is by using a **Function**. A **Function** procedure can return a value; a **Sub** procedure can't.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Outcome 3

The principal command-line scripting language used in a Windows environment is JScript.

1. Writing command-line scripts.

Candidates should be aware that JScript is written in text format, and is organized into statements, blocks consisting of related sets of statements, and comments. Statements can contain variables, immediate data such as strings and numbers and expressions. A JScript statement consists of one or more items and symbols on a line. A new line begins a new statement, but it is a good idea to terminate statements explicitly with a semicolon (;).

Candidates should know that a group of JScript statements surrounded by braces is called a block and that blocks are used, for example, in functions and conditionals.

A single-line JScript comment begins with a pair of forward slashes (//). A multiline comment begins with a forward slash and asterisk (/*), and ends with the reverse (*).

The equals sign (=) is used in JScript to indicate the action of assigning a value, e.g.

```
anInteger = 3;
```

means "Assign the value 3 to the variable anInteger", or "anInteger takes the value 3". A pair of equal signs (==) is used to compare two values to find out whether they are equal.

A JScript expression is something that a person can read as a Boolean or numeric expression. Expressions contain symbols "+" rather than words like "added to". Any valid combination of values, variables, operators, and expressions constitutes an expression.

2. Variables

Candidates should be aware that Variables are used in JScript to store retrieve and manipulate values in scripts. It is considered good practice to declare variables by using the var statement before using them.

```
var total = 3;  
var yes = true;
```

Candidates should know the rules for constructing variable names and be able to use null variables and undefined variables, declare variables implicitly and coerce variables to different types.

3. Data types

JScript has six types of data, numbers, strings, objects, Booleans, null and undefined.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Strings are delineated by single or double quotation marks. A string is also an object in JScript, but it is a special case, with special properties. A string can contain zero or more Unicode characters. When it contains zero, it is called a zero-length string ("").

JScript supports both integer and floating-point numbers. Integers can be positive, 0, or negative; a floating-point number can be written in decimal or scientific notation (IEEE 754). Integers can be represented in base 10 (decimal), base 8 (octal), and base 16 (hexadecimal).

The possible Boolean values are **true** and **false**. In a comparison, any expression that evaluates to 0 is taken to be false, and any statement that evaluates to a number other than 0 is taken to be true.

An undefined value is a value given to a variable after it has been created, but before a value has been assigned to it. A **null** value is one that has no value and means nothing.

4. Operators

Candidates should be aware that JScript has a full range of operators, including arithmetic, logical, bitwise, and assignment operators. There are also a few miscellaneous operators. Operators follow an order of precedence, which can be amended by the use of parentheses.

5. Program flow

Candidates should be aware that JScript provides a range of control structures, the simplest being **if** and **if...else** conditional statements. All conditional tests are Boolean, so the result of any test is **true** or **false**. JScript also supports an implicit conditional form which uses a question mark after the condition to be tested, rather than the word **if** before the condition, and specifies two alternatives, one to be used if the condition is met and one if it is not.

JScript provides several ways to execute a statement or block of statements repeatedly. Repetitive execution is called looping and is typically controlled by a test of some variable, the value of which is changed each time the loop is executed. JScript supports many types of loop: **for** loops, **for...in** loops, **while** loops, **do...while** loops, and **switch** loops.

The **break** statement can be used to stop execution if some (presumably special) condition is met. The **continue** statement can be used to jump immediately to the next iteration, skipping the rest of the code block but updating the counter variable as usual if the loop is a **for** or **for...in** loop.

6. Functions

Candidates should be aware that JScript functions perform actions and can also return results. Sometimes these are the results of calculations or comparisons. Functions combine several operations under one name. A user can write out a set of statements, name it, and then execute the entire set any time it is needed, simply by calling it and passing to it any required information.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Information is passed to a function by enclosing it in parentheses after the name of the function. Items of information passed to a function are called arguments or parameters. Some functions don't take any arguments, some functions take one argument and some take several.

JScript supports both built-in functions and user-defined functions. Built-in functions can be used to handle expressions and special characters and convert strings to numeric values. User-defined functions can be created and used wherever they are needed. A function definition consists of a function statement and a block of JScript statements.

7. Objects

Candidates should be aware that Jscript objects are collections of properties and methods. A method is a function that is a member of an object and a property is a value or set of values (in the form of an array or object) that is a member of an object. JScript supports three kinds of objects: intrinsic objects, user-defined objects and browser objects.

Objects and arrays are handled identically. Any of the members of an object (its properties and methods) can be referred to by name (using the name of the object, followed by a period, followed by the name of the property) or by its array subscript index. Subscript numbering in JScript begins with 0. Subscripts can also be referred to by name.

8. Reserved keywords

Candidates should be aware that JScript has a number of reserved keywords. These words come in three types: JScript reserved keywords, future reserved words and words to avoid.

Outcome 4

1. Creating scripts

Candidates should be aware that WSH provides several different ways of working with scripts. The easiest is to use Notepad to create a script using a single scripting language and save it using the appropriate extension: .vbs for VBScript, or .js for Jscript.

2. Running scripts

Candidates should be aware that scripts are run using the scripting hosts provided by WSH:

WScript: a scripting host with GUI controls for displaying output in pop-up dialog boxes. Used primarily for executing scripts from the desktop.

Cscript: the command-line version of the scripting host. All output from is displayed at the Windows command prompt unless otherwise specified.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

3. Creating batch scripts

Candidates should be aware that multiple types of scripts can be combined in a batch script. With batch scripts, use a single file and are saved with the .wsf file extension. Because batch scripts can use scripts written in multiple scripting languages, the type of scripts being used and other aspects of these scripts must be identified. This is done using markup tags written in XML (Extensible Markup Language).

4. Displaying text strings

Candidates should be aware that the Echo command can be used to display text strings. If CScript is being used in interactive mode, output is written to the command line. If WScript is being used in interactive mode, output is displayed in a pop-up dialog box.

5. Examining script information

Candidates should be aware that scripts may need to check the script host version and the script engine version before running, e.g. if the user system is running the version 4 script engines, version 5 scripts may produce unpredictable results. Other scripting information which may be checked includes the location of the script hosts on the user's system, arguments passed in to the script at start-up, and environment variables set on the system.

Information related to the script hosts (WScript or CScript), the script engine and the current script can be examined using built-in functions in VBScript and JScript. To examine script properties, you use properties of the **WScript** object. Script arguments are placed in a container object called **WshArguments**. Both VBScript and JScript support a property for determining how many arguments were passed in.

6. Working with environment variables

Candidates should be aware that it is possible to access environment variables through **WshEnvironment**, then use the **WshShell.Environment** method to specify which type of environment variables to work with. Environment variables can be divided into four classes:

- **System:** refers to system environment variables
- **User:** refers to user environment variables
- **Volatile:** refers to temporary environment variables
- **Process:** refers to process variables

7. Running programs from within scripts

Candidates should be aware that the **Run()** method of the **WScript.Shell** object allows programs to be run from within scripts. It is possible to:

Higher National Unit specification: support notes (cont)

Unit title: Scripting

- Start Windows applications, such as Microsoft Word, Excel, or PowerPoint
- Run command-line programs, such as shutdown.exe or regedt32.exe
- Run command shell scripts

8. Combining scripts written in different languages.

Candidates should be aware that batch scripts (.WSF files) can be used to combine scripts written in JScript and scripts written in VBScript in the same file. It is possible to call functions in one script from another script and return values to the caller. In VBScript, it is possible to call JScript functions and return values; in JScript, it is possible to call VBScript subroutines to execute a section of code or call VBScript functions and return values.

Outcome 5

1. Input and Output Concepts

Candidates should know that basic I/O is handled through the standard input and standard output streams—much like I/O is handled in most programming languages. Errors are written to the standard error stream. The WScript object has three special properties for working with the input, output, and error streams:

- StdIn: a read-only input stream
- StdOut: a write-only output stream
- StdErr: a write-only error stream

2. Using Input Boxes and Message Boxes

Candidates should be aware that input boxes are only available in VBScript. They can be regarded as customisable dialog boxes that can be used to get input from users. This input can be any kind of text, such as a file path, the user's login name, or a response to a question. Message boxes are only available in VBScript. They can be used to display information and to allow users to make selections. Because message boxes can have customized buttons and icons, they are a bit more complex than input boxes.

3. Error Detection and Handling

Candidates should know that if a user enters the wrong information, the script should handle the error appropriately. Some error-handling techniques are very basic, such as using control loops to ensure that users enter information. Other error-handling techniques are more advanced and usually involve the built-in error-detection capabilities of VBScript and JScript. These include preventing runtime errors from halting execution, checking for and catching errors and examining error codes and descriptions.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

4. File system objects

Candidates should be aware that the **FileSystemObject** is implemented in the scripting runtime library (Srrun.dll) and is an extension of the JScript and VBScript scripting engines rather than a part of the Windows Script Host object model. Many different objects and collections are accessed through the **FileSystemObject**. Objects are containers for related sets of methods and properties. Collections are containers for groups of related items, such as the **Drives** collection, which contains references for all the drives on a particular system. Collections are accessed through the properties and methods of other objects, e.g. the drives on a system could be examined through the **Drives** property of the **FileSystemObject**.

5. Working with folders

Candidates should know how to use scripts to carry out key folder administration tasks, including: viewing folder contents, examining and working with folder properties, checking for and creating folders, deleting, copying, and moving folders and working with special folders

6. Working with files

Candidates should be aware that scripts can be used to use scripts to copy, move, and delete files and to create, read, and write text files. The types of text files can include HTML, XML, scripts, and other types of files containing standard ASCII or Unicode text. Scripts can be used to accomplish administration tasks, including: examining and working with file properties, copying, moving, and deleting files, checking for and creating files and reading and writing files

7. Reading text files

Candidates should know that, prior to working with a file, it needs to be opened for reading, writing, or appending. A text file can only be read if it has been opened in the **ForReading** access mode. Once the file is opened for reading, information can be read in several different ways. It is possible to read the entire contents of the file, read character strings from the file, or read lines of information from the file. It is also possible to skip characters or lines in a file.

8. Writing to a file

Candidates should know that text files can be written using the **ForWriting** and **ForAppending** modes. The access mode determines the initial position of the pointer within the file. With the **ForWriting** mode, the file is initialised, erasing any existing data and the pointer is set at the beginning of the file. With the **ForAppending** mode, the pointer is set to the end of the file and any data written is added to the existing data in the file.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Outcome 6

1. Managing drives

Candidates should know that there are two different ways of working with drives: working with a specific drive, such as the C: drive, or working with drive collections, which are containers for all the local and network drives on a particular system. Most functions that work with drives allow drives to be referenced by drive letter, drive path or network share path. Most network drives have a drive designator associated with them as well as a path. The drive designator for a network drive can be obtained by using the **GetDriveName** method of **FileSystemObject**.

It is useful to test for a drive's existence before trying to work with it. This can be accomplished by using the **DriveExists** method of **FileSystemObject**, which returns **True** if a drive exists. Another common task is obtaining a Drive object, which can then be used to check the drive's properties. A Drive object can be obtained by using the **GetDrive** method of **FileSystemObject**. One of the most useful drive properties is **FreeSpace**, which can be used to keep track of system resources throughout the network.

The easiest way to examine all drives on a system is to use the Drives collection. Network drives allow users and scripts to access remote resources on the network, but network shares aren't automatically available to users or to scripts. A network share must be specifically mapped to a network drive before it is available. In Windows scripts, network drives are mapped using the **MapNetworkDrive** method of the Network object. When access to a network drive is no longer required, the **RemoveNetworkDrive** can be used to disconnect it.

2. Managing network printers

Candidates should be aware that Windows scripts can configure default printers, as well as add and remove network printers. A network printer is a shared printer that is accessible to other systems on the network. The default printer is used whenever a user prints a document and doesn't select a specific destination printer. The default printer can be set by using the **SetDefaultPrinter** method of the Network object. This method automatically updates the user's profile to use the default printer in the current session as well as subsequent sessions. Windows scripts manage connections to network printers much like they manage connections to network drives. Printer connections can be mapped by using **AddPrinterConnection** or **AddWindowsPrinterConnection** and removed by using **RemovePrinterConnection**.

3. Managing shortcuts and menus

Candidates should know that menus, desktops, and startup applications are all configured with shortcuts and it is the location of the shortcut that determines how the shortcut is used. For example, a menu option for a user can be added by adding a shortcut to the user's Programs or Start folder. These shortcuts then appear on the user's menu. Scripts can also be used to create, update and delete menus.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

4. Adding and removing startup applications

Candidates should know that applications that should be started after a user logs on can be set up by creating shortcuts in the **AllUsersStartup** and **Startup** folders. The **AllUsersStartup** folder sets startup applications for all users that log onto a system. The **Startup** folder sets startup applications for the current user.

5. Working with system data.

Candidates should know that Windows Script Host can be used to manage the Windows Registry. The registry stores configuration information for the operating system, applications, services, and more. By examining and changing registry information in scripts, a system can be reconfigured so that it runs exactly as required. Windows scripts can be used to read, write, and delete registry entries, and to reconfigure network services, such as DHCP and WINS.

Because the registry is essential to the proper operation of the operating system, changes should only be made when it is known how they will affect the system. Improperly modifying the Windows Registry can cause serious problems and if the registry gets corrupted, the operating system may need to be reinstalled. Registry scripts must always be double-checked before running them, to ensure that they perform exactly as intended.

6. Writing to event logs

Candidates should know that Windows Script Host can be used to manage Windows event logs. The event logs track essential processes on a system and can also be used in auditing system activity. By examining event logs through scripts, system activity can be analysed and a system can be monitored for problems. Windows event logs can be used to track system processes, to troubleshoot system problems and to monitor system security. Windows 2000/2003 servers and workstations have the following logs:

- **Application Log:** tracks events logged by applications, such as SQL Server.
- **Security Log:** tracks events set for auditing with local or global group policies. Only authorized users can access security logs.
- **System Log:** tracks events logged by the operating system or its components, such as WINS or DHCP.
- **Directory Service:** tracks events logged by Active Directory.
- **DNS Server:** tracks DNS queries, responses, and other DNS activities.
- **File Replication Service:** tracks file replication activities on the system.

Event logs can be used to monitor critical systems and to create customised reports based on event logs entries. The **EventLog** method makes writing to event logs fairly easy.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

7. Reading event logs

Candidates should know that there isn't a simple method of reading event logs. This is because event logs have a complex structure which requires a tool that can search the event logs for relevant information, rather than a tool that simply reads the events. The built-in capabilities of VBScript and JScript can be used to create log-searching and extraction routines, but this isn't really necessary as Dumpel can handle all of the dirty work. Dumpel is a resource kit utility designed to help analyse event logs.

Higher National Unit specification: support notes (cont)

Unit title: Scripting

Guidance on the delivery and assessment of this Unit

This Unit is likely to form part of a group award which is primarily designed to provide candidates with technical or professional knowledge and skills related to a specific occupational area. It is highly technical in content and should not be adopted by group awards in other areas or delivered as a stand-alone Unit without careful consideration of its appropriateness. It is a Unit which candidates are likely to find accessible at an introductory level; it is suggested that it be delivered as part of a first-year or second-year HNC/HND program in Computing or a related area. It should be delivered in tandem with other Computing Units and opportunities for teaching and assessment integration explored.

To minimise assessment overhead, sets of restricted-response questions are used to provide evidence of candidates' knowledge for all Outcomes. It is suggested that multiple-choice questions can be used as the preferred assessment method – as well as reducing the time required for assessment and marking, these reduce the need for candidates to memorise details and encourage understanding. The number of questions which must be answered correctly corresponds to 70% of those set in each case.

The restricted-response questions should be supplemented by collection of short scripts, illustrating particular Scripting features. These short scripts may be combined to carry out meaningful tasks, but this is not mandatory.

Open learning

If this Unit is delivered by open or distance learning methods, additional planning and resources may be required for candidate support, assessment and quality assurance.

A combination of new and traditional authentication tools may have to be devised for assessment and re-assessment purposes.

For further information and advice, please see *Assessment and Quality Assurance for Open and Distance Learning* (SQA, February 2001 — publication code A1030).

Special needs

This Unit specification is intended to ensure that there are no artificial barriers to learning or assessment. Special needs of individual candidates should be taken into account when planning learning experiences, selecting assessment instruments or considering special alternative Outcomes for Units. For example, some candidates may require a longer period for the single assessment or may require that it be split into more than one event. For information on these, please refer to the SQA document *Guidance on Special Assessment and Certification Arrangements for Candidates with Special Needs and Candidates for Whom English is an Additional Language* (SQA, 2000).

General information for candidates

Unit title: Scripting

This is a 2-credit Unit at Level 7 intended for candidates undertaking a Computing or IT-related qualification who require an understanding of Scripting. It is designed to develop an understanding of the issues involved in writing scripts in GUI or command-line based scripting languages. On completion of the Unit you should be able to:

- Describe the scripting environment.
- Describe GUI scripting concepts.
- Describe command-line scripting concepts.
- Write simple scripts.
- Handle I/O and files.
- Write complex scripts.

In the first part of the course, you will study scripting concepts, including operating system script host components, using and running scripts and operating system script host architecture.

The second section covers GUI scripting concepts, including data types, variables and constants, operators, program flow and procedures.

The third section covers command-line concepts, including writing scripts, variables, data types, operators, program flow, functions, objects and reserved words.

The fourth section covers writing simple scripts, including creating scripts, running scripts, creating batch scripts, displaying text strings, examining script information, working with environment variables, running programs from within scripts and combining scripts written in different languages.

The fifth section covers handling I/O and files, including input and output concepts, using input boxes and message boxes, error detection and handling, file system objects, working with folders, working with files, reading text files and writing to a file.

The final section covers writing complex scripts, including managing drives, managing network printers, managing shortcuts and menus, adding and removing startup applications, working with system data, writing to event logs and reading event logs.

There will be a closed-book multiple-choice assessment covering all outcomes. You will be presented with 50 questions and expected to answer 70% of these correctly. You will also be expected to produce a collection of short scripts, illustrating particular scripting features.

You must satisfy the requirements for these assessments in order to achieve the Unit.