Course Support Notes

National 4 Computing Science Course Support Notes

Please refer to the note of changes at the end of this document for details of changes from previous version (where applicable).

Contents

Introduction

These support notes are not mandatory. They provide advice and guidance on approaches to delivering and assessing the National 4 Computing Science Course. They are intended for teachers and lecturers who are delivering the Course and its Units. They should be read in conjunction with the *Course Specification*, the *Added Value Unit Specification*, and the *Unit Specification*s for the Units in the Course.

General guidance on the Course

Aims

As stated in the *Course Specification*, the aims of the Course are to enable learners to:

- introduce and develop aspects of computational thinking across a range of contemporary contexts
- develop knowledge and understanding of key facts and ideas in computing science
- apply skills and knowledge in analysis, design, implementation and testing to a range of digital solutions
- communicate computing concepts clearly and concisely using appropriate terminology
- develop an understanding of the impact of computing science in changing and influencing our environment and society

Related to these aims, and underlying the study of computing science, are a number of unifying themes, including technological progress and trends, the relationship between software, hardware and system performance and information representation and transfer as a core component of any computation. These are used to explore a variety of specialist areas through practical and investigative tasks.

This Course will also give learners the opportunity to develop thinking skills and skills in numeracy, employability, enterprise and citizenship.

Progression into this Course

Entry to this Course is at the discretion of the centre. However, learners would normally be expected to have attained some relevant skills and knowledge through prior experience.

Skills and knowledge developed through any of the following, while not mandatory, are likely to be helpful as a basis for further learning for this Course:

Other SQA qualifications

- National 3 Computing Science Course or relevant component Units
- National 3 Numeracy

Experiences and outcomes

New National Courses have been designed to draw on and build on the curriculum experiences and outcomes as appropriate. Qualifications developed for the senior phase of secondary education are benchmarked against SCQF levels. SCQF level 4 and the curriculum level 4 are broadly equivalent in terms of level of demand although qualifications at SCQF level 4 will be more specific to allow for more specialist study of subjects.

Learners who have completed Curriculum for Excellence experiences and outcomes will also find these an appropriate basis for doing the Course.

In this Course, any of the following may be relevant:

- Having investigated a current trend of technological advance in Scotland or beyond, I can debate the short- and long-term possibilities of the technological development becoming a reality (TCH 4-01a)
- I can debate the possible future impact of new and emerging technologies on economic prosperity and the environment. (TCH 4-01c)
- By discussing the business, environmental, ethical and social implications of computer technology, I can begin to gain an understanding of the need for sustainability and accessibility. (TCH 4-05a)
- I can compare different forms of security software to gain knowledge and understanding of their functions in protecting contemporary technologies. (TCH 4-08b)
- I can integrate different media to create a digital solution which allows interaction and collaboration with others. (TCH 4-08c)
- Through research, I can gain knowledge of computer systems or emerging technologies to understand their differing features and consider their suitability for the world of work. (TCH 4-08d)
- By learning the basic principles of a programming language or control technology, I can design a solution to a scenario, implement it and evaluate its success. (TCH 4-09a)
- I can create graphics and animations using appropriate software which utilise my skills and knowledge of the application. (TCH 4-09b)
- I can use features of software to create my own animation which can then be used to create an animated sequence. (TCH 4-09c)

Other experience
Learners may have relevant skills and knowledge gained through other prior learning, life and work experiences.

Skills, knowledge and understanding covered in this Course

This section provides further advice and guidance about skills, knowledge and understanding that could be included in the Course.

Note: teachers and lecturers should refer to the *Added Value Unit Specification* for mandatory information about the skills, knowledge and understanding to be covered in this Course.

The mandatory skills may be developed throughout the Course. The table below shows where there are significant opportunities to develop these in individual Units.

Mandatory skills and knowledge	**Software Design and Development**	**Information System Design and Development**	**Added Value**
applying, with guidance, aspects of computational thinking across a range of straightforward contexts	✓	✓	✓
analysing, with guidance, straightforward problems within computing science across a range of contemporary contexts			✓
designing, implementing and testing, with guidance, digital solutions (including computer programs) to straightforward problems across a range of contemporary contexts	✓	✓	✓
developing skills in computer programming and the ability to communicate how a program works by being able to read and interpret code	✓		✓
communicating basic understanding of key concepts related to software design and development clearly and concisely using appropriate terminology	✓		✓
communicating basic understanding of key concepts related to information system design and development clearly and concisely using appropriate terminology		✓	✓
basic knowledge of the impact of contemporary technologies on the environment and society	✓	✓	✓
applying basic computing and information concepts and techniques to create solutions	✓	✓	✓

Teachers/lecturers should ensure that learners are fully aware of the wide range of skills, knowledge and understanding that they are developing in the Units and Course as a whole.

It is also important to highlight any transferable learning that is taking place which supports the development of skills for learning, skills for life and skills for work.

Progression from this Course

This Course or its Units may provide progression to:

- National 5 Computing Science Course
- National Certificate Group Awards in computing, IT and related disciplines
- other technological courses at National 4
- employment, apprenticeships and/or training in IT and related fields

and ultimately, for some, to:

- Higher and Advanced Higher Computing Science Courses
- a range of computing-related Higher National Diplomas (HNDs)
- degrees in Computing, IT and related disciplines
- careers in computing, IT and related disciplines

Hierarchies

Hierarchy is the term used to describe Courses and Units which form a structured progression involving two or more SCQF levels.

It is important that any content in a Course and/or Unit at one particular SCQF level is not repeated (unless required for consolidation) if a learner progresses to the next level of the hierarchy. The skills and knowledge should be able to be applied to new content and contexts to enrich the learning experience. This is for centres to manage.

The Course is designed in hierarchy with the corresponding Courses at SCQF levels 5 and 6 (National 5 and Higher). The Computing Science Courses at all three levels have the same structure of two Units with corresponding titles.

Appendix 2 contains a table showing the relationship between the mandatory National 4 and National 5 knowledge and understanding. This may be useful for:

- designing and planning learning activities for National 4/National 5
- ensuring seamless progression between levels
- identifying important prior learning for learners at National 4

Teachers should also refer to the Outcomes and Assessment Standards for each level when planning delivery.

Further advice on delivery to a group including National 4 and National 5 learners is given in the next section of these support notes, with additional detailed guidance in the *Unit Support Notes*.

Approaches to learning and teaching

Computing Science, like all new and revised National Courses, has been developed to reflect Curriculum for Excellence values, purposes and principles.

The approach to learning and teaching developed by individual centres should reflect these principles. Learners should be encouraged to participate fully in active learning and practical activities by working together, talking, listening, reading or reflecting on a topic while the teacher acts as a facilitator.

An appropriate balance of teaching methodologies should be used in the delivery of the Course and a variety of active learning approaches is encouraged, including the following:

Activity-based learning
Whole-class, direct teaching opportunities should be balanced by activity-based learning on practical tasks. An investigatory approach is encouraged, with learners actively involved in developing their skills, knowledge and understanding by investigating a range of 'real-life' and relevant problems and solutions related to areas of study. Learning should be supported by appropriate practical activities, so that skills are developed simultaneously with knowledge and understanding.

Group work
Practical activities and investigations lend themselves to group work, and this should be encouraged. Working within a group will encourage learners to collaborate and work cooperatively with others. Learners engaged in group work strategies capitalise on one another's knowledge, resources and skills in questioning, investigating, evaluating and presenting ideas to one another. While "working as a team" is not specifically identified as one of the skills for life, learning and work for this Course and therefore not assessed, it is a fundamental aspect of working in the IT and related industries and so should be encouraged and developed by teachers.

Problem-Based Learning
Problem-Based Learning (PBL) is another strategy which will support a learner's progress through this Course. This method may be best utilised at the end of an Outcome or a topic where additional challenge is required to ensure learners are secure in their knowledge and understanding, and to develop the ability to apply knowledge and skills in less familiar contexts. Learning through PBL develops a learner's problem solving, decision-making, investigative skills, creative thinking, team working and evaluative skills.

Computational thinking
Computational thinking is recognised as a key skill set for all 21st century learners — whether they intend to continue with computing science or not. It involves a set of problem-solving skills and techniques used by software developers to write programs.

There are various ways of defining computational thinking. One useful structure is to group these problem-solving skills and techniques under five broad headings:

- **Abstraction:** seeing a problem and its solution at many levels of detail and generalising the information that is necessary. Abstraction allows us to represent an idea or a process in general terms (eg variables) so that we can use it to solve other problems that are similar in nature.
- **Algorithms:** the ability to develop a step-by-step strategy for solving a problem. Algorithm design is often based on the decomposition of a problem and the identification of patterns that help to solve the problem. In computing science as well as in mathematics, algorithms are often written abstractly, utilising variables in place of specific numbers.
- **Decomposition:** breaking down a task so that we can clearly explain a process to another person — or to a computer. Decomposing a problem frequently leads to pattern recognition and generalisation/abstraction, and thus the ability to design an algorithm.
- **Pattern recognition:** the ability to notice similarities or common differences that will help us make predictions or lead us to shortcuts. Pattern recognition is frequently the basis for solving problems and designing algorithms.
- **Generalisation:** realising that a solution to one problem may be used to solve a whole range of related problems.

Underpinning all of these concepts is the idea that computers are **deterministic**: they do exactly what you tell them to do. The corollary of this, of course, is that they can be understood.

Whilst computational thinking can be a component of many subjects, Computing Science is particularly well placed to deliver it. Teachers are encouraged to emphasise, exemplify and make these aspects of computational thinking explicit (at an appropriate level) wherever there are opportunities to do so throughout the teaching and learning of this Course and its Units.

Using online and outside resources

Throughout the teaching of this Course, the stimulation of learners' interest and curiosity should be a prime objective. Engagement with outside agencies or industry professionals can greatly enhance the learning process. Online resources, such as those listed in the individual Unit Support Notes may provide a valuable addition to teaching and learning activities, encouraging research, collation and storage of information and evaluation of these materials. The use of interactive multimedia learning resources, online quizzes, and web-based software can also be used to support activity-based approaches.

Assessment activities, used to support learning, may be usefully blended with learning activities throughout the Course, for example by:

- sharing learning intentions/success criteria
- using assessment information to set learning targets and next steps
- adapting teaching and learning activities based on assessment information
- boosting learners' confidence by providing supportive feedback

Self- and peer-assessment techniques should be encouraged wherever appropriate.

Working towards Units and Course
Learning and teaching activities should be designed to develop both:

- skills and knowledge to the standard required by **each Unit** and to the level defined by the associated Outcomes and Assessment Standards
- ability to apply the breadth of knowledge, understanding and skills required to complete the **Added Value Unit** successfully

Meeting the needs of all learners
Within any class, each learner will have individual strengths and areas for improvement.

For example, within a National 4 class, there may be learners capable of achieving National 5 standards in some aspects of the Course. Where possible, they should be given the opportunity to do so.

Teachers need to consider both the Outcomes and Assessment Standards, and the table of content in Appendix 2 of these notes, to identify the differences between National 4 and National 5.

The difference between National 4 and National 5 is defined in terms of a higher level of skill. For example, in the *Information System Design and Development* Unit, Outcome 1 requires National 4 learners to 'develop simple information systems using appropriate development tools', by creating a structure with links, integrating different media types and identifying and rectifying errors, while at National 5 in the similar Outcome 1 learners are also required to create a user interface and write or edit simple code.

When delivering this Course to a group of learners, with some learners working towards National 4 and others towards National 5, it may be useful for teachers to identify activities covering common knowledge and skills for all learners, and additional activities for National 5 learners. This is particularly appropriate where the National 5 learners have come directly from the broad general education without previously studying National 4.

However, where National 5 learners have studied National 4 in a previous year, it is important to provide them with new and different contexts for learning to avoid de-motivation.

Learning about Scotland and Scottish culture will enrich the learners' learning experience and help them to develop the skills for learning, life and work they will need to prepare them for taking their place in a diverse, inclusive and participative Scotland and beyond. Where there are opportunities to contextualise approaches to learning and teaching to Scottish contexts, teachers and lecturers should consider this.

Sequence of delivery of Units
The sequence of delivery of the Units within the National 4 Computing Science Course is a matter of professional judgement and is at the discretion of the centre.

Units could be delivered in sequence or in parallel. One approach might be to alternate delivery of the Units' content to give variety to teaching and learning activities, eg programming in a software development environment alternating with developing web pages, but other possibilities exist.

Whether the decision is taken to deliver Units sequentially or in parallel, it would be good practice to complete the two other Units before attempting the Added Value Unit. This approach will give learners the opportunity to develop the skills and knowledge necessary to enable them to successfully attempt the added value assignment. However, it may be possible to begin work on the Added Value Unit at an earlier stage, but only where it is clear that learners have already gained the required skills and knowledge.

Advice on distribution of time
The distribution of time between the Units is a matter for professional judgement and is entirely at the discretion the centre.

Within the framework for completing the Added Value Unit (*Computing Science Assignment*) time will be required for:

- preparation for the assignment, which could include considering exemplar assignments and practising the application and integration of skills
- carrying out the stages of the assignment, with teacher guidance and support
- assessing the process and completed solution
- providing opportunities for re-assessment if required
- completing a record of progress

Resources
Centres may find that existing hardware and software within the computing science classroom provides all that is required to deliver the Course. The resources required are summarised below:

- internet-enabled computers and a digital projector
- access to software development tools (appropriate software development environments)
- access to application development software and tools (macro editors, applications that support data handling, presentation, group work, animation, video, graphics and text)
- web development tools (script enabled browsers)
- digital media devices (scanners, digital cameras, camcorders, etc)

Developing skills for learning, skills for life and skills for work

Guidance on the development of skills for life, skills for learning and skills for work is to be found in the *Unit Support Notes* for each of the component Units.

Approaches to assessment

See the *Unit Support Notes* for guidance on approaches to assessment of the Units of the Course.

Added value

Courses from National 4 to Advanced Higher include assessment of added value.

At National 4 the added value will be assessed in the Added Value Unit.

Information given in the *Course Specification* about the assessment of added value is mandatory. Full details of assessment of the assignment are included in the *Added Value Unit Specification.*

The Added Value Unit will assess the application of skills and knowledge which learners will have developed through the other Units. Evidence for this Unit will be generated through an assignment in which learners will be required to solve an appropriately challenging computing science problem.

Combining assessment across Units

If an integrated approach to Course delivery is chosen, then there may be opportunities for combining assessment across Units.

Equality and inclusion

The requirement to develop practical skills involving the use of equipment and tools may present challenges for learners with physical or visual impairment. In such cases, reasonable adjustments may be appropriate, including (for example) the use of adapted equipment or alternative assistive technologies. This is for both candidates and centres to consider.

It is recognised that centres have their own duties under equality and other legislation and policy initiatives. The guidance given in these *Course Support Notes* is designed to sit alongside these duties but is specific to the delivery and assessment of the Course.

It is important that centres are aware of and understand SQA's assessment arrangements for disabled learners, and those with additional support needs, when making requests for adjustments to published assessment arrangements. Centres will find more guidance on this in the series of publications on Assessment Arrangements on SQA's website: www.sqa.org.uk/sqa/14977.html.

Appendix 1: Reference documents

The following reference documents will provide useful information and background.

- Assessment Arrangements (for disabled candidates and/or those with additional support needs) — various publications are available on SQA's website at: www.sqa.org.uk/sqa//14977.html.
- British Computer Society, Glossary of Computing and IT, 12th edition
- Building the Curriculum 4: Skills for learning, skills for life and skills for work
- Building the Curriculum 5: A framework for assessment
- Course Specifications
- Design Principles for National Courses
- Guide to Assessment (June 2008)
- Overview of Qualification Reports
- Principles and practice papers for curriculum areas
- SCQF Handbook: User Guide (published 2009) and SCQF level descriptors (to be reviewed during 2011 to 2012): www.sqa.org.uk/sqa/4595.html
- SQA Skills Framework: Skills for Learning, Skills for Life and Skills for Work
- Skills for Learning, Skills for Life and Skills for Work: Using the Curriculum Tool

Appendix 2: Comparison of National 4 and National 5

This table shows the relationship between the mandatory National 4 and National 5 knowledge and understanding. This table may be useful for:

- designing and planning learning activities for multi-level National 4/National 5 classes
- ensuring seamless progression between levels
- identifying important prior learning for learners at National 5

Teachers should also refer to the Outcomes and Assessment Standards for each level when planning delivery.

NB: Where similar topics are covered at both levels, the Outcomes, Assessment Standards and Evidence Requirements distinguish the level of treatment.

Software Design and Development		
Topic	**National 4**	**National 5**
Computational constructs	♦ Exemplification and implementation of the following constructs: — expressions to assign values to variables — expressions to return values using arithmetic operations (+, -, *, /, ^) — execution of lines of code in sequence demonstrating input — process — output — use of selection constructs including simple conditional statements — iteration and repetition using fixed and conditional loops	♦ Description, exemplification and implementation of the following constructs: — expressions to assign values to variables — expressions to return values using arithmetic operations (+, -, *, /, ^, mod) — expressions to concatenate strings and arrays using the & operator — use of selection constructs including simple and complex conditional statements using logical operators (AND, OR, NOT) — iteration and repetition using fixed and conditional loops — pre-defined functions (with parameters) including Random, Integer and Round
Data types and structures	— String — numeric (integer) variables — graphical objects	♦ Description, implementation and exemplification of the following data types and

		structures: — character — String — numeric (integer and real) variables — Boolean variables — 1-D arrays
Testing and documenting solutions	♦ Normal, extreme and exceptional test data ♦ Readability of code (internal commentary, meaningful variable names)	♦ Description, identification, exemplification and implementation of normal, extreme and exceptional test data ♦ Description and identification of syntax, execution and logic errors ♦ Description, identification and exemplification of the readability of code including: — internal commentary — meaningful identifiers — indentation — white space
Algorithm specification		♦ Description, exemplification and implementation of standard algorithms, including: — input validation
Design notations (also applies in information system design and development)	♦ Graphical to illustrate selection and iteration	♦ Description and identification of structure diagrams, flowcharts and pseudocode to solve problems ♦ Exemplification of pseudocode to solve problems
Low-level operations and computer architecture	Use of binary to represent and store: ♦ positive integers ♦ characters ♦ instructions (machine code) Units of storage (bit, byte, Kb, Mb, Gb, Tb, Pb)	♦ Explanation of the need to Translate high-level program code to binary (machine code) ♦ Comparison of interpreters and compilers ♦ Description and exemplification of the use of binary to represent positive integers ♦ Conversion from Binary to decimal and vice-versa ♦ Description of floating point

		representation of real numbers using the terms mantissa and exponent ♦ Description of ASCII code (7-bit) used to represent characters ♦ Description of the vector graphics method of graphic representation ♦ Description of the bit-mapped method of graphics representation ♦ Describe the purpose of the Basic computer architecture components and how they are linked together including: — processor (registers, ALU, control unit) — memory — buses (data and address) — interfaces
Contemporary developments		♦ Exemplification of trends in the development of: — software development languages — software development environments — their editing features — high-level code translation and execution

Information System Design and Development		
The following mandatory generic concepts and vocabulary may be applicable to a range of information systems types and contexts (including databases, websites, games, mobile applications, kiosk systems).		
Topic	**National 4**	**National 5**
Structures and links (database)	♦ database structure: field, record, file ♦ field types (text, number, date, time, graphic, calculated) ♦ database operations (search, sort)	♦ Implementation of a relational database with two linked data tables ♦ Advantages of relational database over flat-file databases ♦ Description, exemplification and implementation of primary keys and foreign keys ♦ Description and

		exemplification of field types (text, number, date, time, graphic, object, link, Boolean) ♦ Description and exemplification of validation including: — presence check — restricted choice — field length — range ♦ Description and exemplification of database operations search, sort (on multiple fields) and calculations ♦ Description, exemplification and implementation of good design to avoid data duplication and modification errors (insert, delete, update)
Structures and links (web-based)	♦ website, page, URL ♦ hyperlink	♦ Description of website, page, URL in relation to a web-based information system ♦ Description and implementation of hyperlinks (internal, external), relative and absolute addressing ♦ Understand the need for, and exemplify, simple navigation within web-based information systems ♦ Description and features of web browsers and search engines
User interface (also applies in software design and development)		♦ Description of requirements for a good user interface including: — visual layout — navigation — selection — consistency — interactivity — readability
Media types	Sound, graphics, video, text	♦ Description of Standard file formats and their benefits ♦ Know a range of standard

		file formats for different media types including: — the text standard file formats txt, rtf — the audio standard file formats wav, mp3 — the graphic standard file formats jpeg, gif, png, svg — the video standard file formats mp4, avi ♦ Describe and exemplify the factors affecting file size and quality, including resolution, colour depth, sampling rate ♦ Calculation of file size for colour bitmap ♦ Description of the need for compression
Coding		♦ Description and identification of coding to create and modify information systems including: — JavaScript mouse events ♦ Description, exemplification and implementation of coding to create and modify information systems including the use of HTML with the tags for: — document — links — graphics
Testing		♦ Description and exemplification of testing information systems including: — links and navigation work correctly — matches user interface design — media such as text, graphics and video display correctly
Purpose, features, functionality, users	Simple descriptions of main features and functionality	♦ Description of purpose of an information system ♦ Description of the features and functions of an information system

		♦ Description of types of users of information systems including: — expert — novice ♦ Description of age-range of users of information systems
Technical implementation (hardware requirements)	♦ input and output devices ♦ processor clock speed (Hz) ♦ memory (RAM, ROM)	♦ Description and exemplification of the appropriate type of hardware required for a specific information system including: — input and output devices — processor type and speed (Hz) — memory capacity (RAM)
Technical implementation (software requirements)	♦ operating system platform required	♦ Describe the purpose of an operating system including: — controlling peripherals — running software — HCI ♦ Understand the features of web browsers including: — OS support — privacy modes — Ad filtering — page zooming ♦ Description and exemplification of the appropriate type of software required for a specific information system including: — type of application — operating system
Technical implementation (storage)	Storage devices: ♦ built-in, external, portable ♦ magnetic, optical ♦ capacity, speed ♦ rewritable, read-only	♦ Comparison of local versus cloud storage ♦ Comparison of built-in versus portable storage ♦ Comparison of different interface types and their data transfer speeds including: — Firewire — USB — bandwidth ♦ Description of different types of storage devices

		and their media in terms of functionality and capacity (in appropriate units) including: — magnetic — optical — solid state ♦ Description and exemplification of the appropriate type of storage required for a specific information system including: — type of device — capacity — interface type
Technical implementation (networking/ connectivity)	♦ stand-alone or networked ♦ LAN/internet ♦ wired/wireless	♦ Description and comparison of the following transmission media in relation to data speeds and ease of use: — wired — optical — wireless ♦ Description and exemplification of hardware required for network connectivity including: — Network Interface Card — router — hub — switch ♦ Description and exemplification of the appropriate type of network connection required for a specific information system including: — hardware — transmission media
Security risks	♦ viruses, worms, Trojans ♦ hacking	♦ Description and identification of the following security risks: — phishing — keylogging (software and hardware) — virus — online fraud — identity theft
Security precautions		♦ Description and exemplification of anti-virus software

		♦ Description and exemplification of good practice in passwords settings ♦ Description and exemplification of biometrics including: — retina scanning — finger prints — palm prints — face recognition ♦ Description and exemplification of firewalls
Legal implications		♦ Description, identification and implications for individuals and businesses of the Computer Misuse Act including: — use of software and hardware to access data unlawfully — impairing of operation of computer systems ♦ Description, identification and implications for individuals and businesses of the Data Protection Act including: — data in electronic transmission — prior consent of data subject — export of data ♦ Description, identification and implications for individuals and businesses of the Copyright, Designs and Patents Act (plagiarism) including: — copyright of computer software — software piracy — web content — text, graphics, video, audio ♦ Description, identification and implications for individuals and businesses of the Communication Acts including: — post of offensive information on social network sites — send offensive, indecent

		or threatening messages on a public electronic communications network — use networks without permission
Environmental impact		♦ Description and implications of the impact of the energy use of computer systems and how it could be reduced including: — settings on monitors — power down settings — leaving computers on stand-by ♦ Description and exemplification of the correct ways to dispose of IT equipment including: — recycle individual components appropriately — extraction of dangerous elements — re-use of systems for other uses

Administrative information

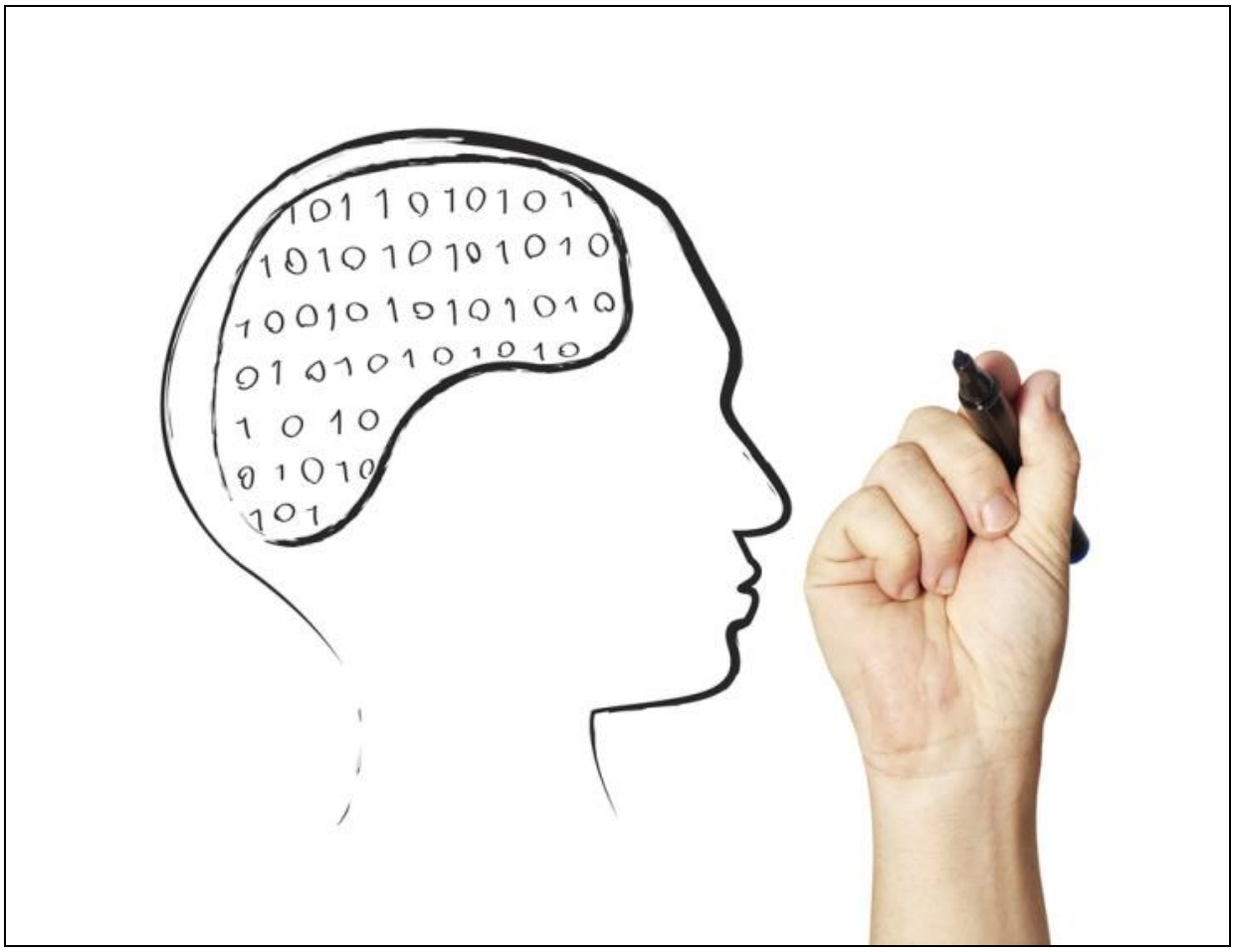
Published: May 2016 (version 2.1)

History of changes to Course Support Notes

Version	Description of change	Authorised by	Date
1.1	Clarifications to mandatory content in Course Support Notes; minor changes to *Software Design and Development* Unit Support Notes; addition of 'storage' to Assessment Standard 2.2 in *Information System Design and Development* Unit Support Notes.	Qualifications Development Manager	July 2013
2.0	Changes made to 'Approaches to learning and teaching' section to reflect removal of Outcome 3.	Qualifications Manager	June 2015
2.1	Changes made to 'Appendix 2: Comparison of National 4 and National 5' to reflect the changes already made to the National 5 Course Assessment Specification.	Qualifications Manager	May 2016

Unit Support Notes

Unit Support Notes — Software Design and Development (National 4)



Please refer to the note of changes at the end of this document for details of changes from previous version (where applicable).

Introduction

These support notes are not mandatory. They provide advice and guidance on approaches to delivering and assessing the *Software Design and Development* (National 4) Unit. They are intended for teachers and lecturers who are delivering this Unit. They should be read in conjunction with:

- the Unit Specification
- the Course Specification
- the Added Value Unit Specification
- the Course Support Notes
- appropriate assessment support materials

General guidance on the Unit

Aims

The purpose of this Unit, as stated in the *Unit Specification*, is for the learner to develop basic knowledge, understanding and practical problem-solving skills in software development through appropriate software development environments. Learners will develop basic computational thinking and programming skills by implementing practical solutions and explaining how these programs work. They will also develop an understanding of how data and instructions are stored in binary form, how programming underpins all computer applications, and an awareness of the impact of contemporary software-based applications on society or the environment.

This Unit will give learners the opportunity to develop their thinking skills as well as their skills in numeracy, employability, enterprise and citizenship.

This Unit can be delivered:

- as a stand-alone Unit
- as part of the National 4 Computing Science Course

Progression into this Unit

Entry to this Unit is at the discretion of the centre. However, learners would normally be expected to have attained some useful skills and knowledge from prior learning, such as:

- National 3 Computing Science Course or relevant component Units
- National 3 Numeracy

Learners may also have gained relevant skills and knowledge through other education systems or from their own interests and informal learning.

Skills, knowledge and understanding covered in this Unit

Information about skills, knowledge and understanding is given in the National 4 Computing Science *Course Support Notes*.

If the Unit is being delivered as part of the National 4 Computing Science Course, the teacher should refer to the 'Further mandatory information on Course coverage' section in the *Added Value Unit Specification* for detailed content.

If this Unit is being delivered on a free-standing basis, teachers are free to select the skills, knowledge, understanding and contexts which are most appropriate for delivery in their centres.

Progression from this Unit

On successful completion of this Unit, the following Units and Courses provide appropriate progression pathways for learners:

- Software Design and Development (National 5) Unit
- National 5 Computing Science Course
- National Certificate Group Awards in a range of Computing, IT and related disciplines
- National Progression Awards in Digital Media
- other technological Courses at National 4
- employment, apprenticeships and/or training in Computing, IT and related fields

Approaches to learning, teaching and assessment

Learning and teaching
The Unit is designed to provide flexibility, personalisation and choice for both the learner and the teacher.

Learning and teaching activities should be designed to stimulate learners' interest, and to develop skills and knowledge to the standard required by the Outcomes and to the level defined by the associated Assessment Standards. Learning should be supported by appropriate practical activities so that skills are developed simultaneously with knowledge and understanding.

An investigatory approach is encouraged, with learners actively involved in developing their skills, knowledge and understanding by investigating a range of real-life and relevant software development environments, problems and solutions.

The Unit and Course Specifications define the skills and knowledge required, but leave complete freedom to the teacher and learner to select interesting contexts and environments in which to develop these.

Practical activities and investigations lend themselves readily to group work, and this should be encouraged. Individual, paired or group problem-solving tasks could be related to authentic and relevant contexts.

The National 4 Computing Science *Course Support Notes* provide further broad guidance on approaches to learning and teaching which may apply to all of the Units of the Course and should be read before delivering this Unit.

When delivering the Unit as part of the National 4 Computing Science Course, reference should be made to the appropriate content statements within the *Added Value Unit Specification* to ensure the required breadth of knowledge is covered.

Sequence of delivery of Outcomes
The sequence of delivery of the Outcomes is a matter of professional judgement and is entirely at the discretion of the centre. Some suggested approaches are listed below.

Outcome 1 and Outcome 2 simultaneously
Teachers may consider it good practice to combine the development of knowledge and understanding of how programs work at the same time as learners progress through the practical tasks involved in developing their own programs.

For example, when learners are selecting and using appropriate constructs and variables when developing their programs, they will need to understand the purpose and function of these constructs and the purpose and roles of variables. At the same time they could be exploring how the data and instructions are then stored by a computer system. As learners develop their programming skills by selecting and using appropriate constructs and assigning values to variables, they will also develop a clear understanding of how these constructs and

variables work and what they can be used for. This will enable them to read and explain the code in their programs.

Outcome 1 before Outcome 2
It is possible to explain how simple software development programs work, and meet the requirements of Outcome 1 before carrying out the practical task of developing short programs in Outcome 2.

For example, learners could gain an understanding of the basic concepts involved in software development before creating their own programs. They could select different programs and then find out the following:

- What is the purpose of the programming constructs?
- How do they work?
- What is the purpose of different variable types

Once the learners have gained a sound understanding of the purpose of a range of programming constructs and variable types, they should be well placed to develop their own simple programs in their chosen software development environment, such as games, mobile application or text-based. They should be able to apply their knowledge gained in Outcome 1 to select and use the appropriate programming constructs to produce a complete, working program.

Outcome 2 before Outcome 1
It is possible to develop the learner's practical skills in developing short programs and meet the requirements of Outcome 2 before moving to formally develop the knowledge and understanding associated with explaining how programs work in Outcome 1.

For example, learners would develop their programming skills in a software development environment by learning the functions of certain programming constructs and using a range of simple data types. They would be coding solutions to problems using these constructs and data types and building up valuable experience in programming.

These activities would allow the learners to fully understand the purpose and function of certain constructs and data types and how data and instructions are stored within programs. Consequently, when learners are then asked to read and explain sections of code or determine the value of a variable within a loop construct within a program in Outcome 1, they will have the necessary knowledge and skills to do this from their experience in coding in Outcome 2.

Meeting the needs of all learners
When teaching this Unit to a class, with some learners working towards National 4 and others towards National 5, it may be useful for teachers to identify activities covering common knowledge and skills for all learners, and additional activities for National 5 learners. For example:

- Computational constructs

When learners are developing their skills and knowledge in computational constructs, both National 4 and National 5 learners should undertake tasks where they have to:

  - use expressions to assign values to variables

- execute lines of code in sequence demonstrating input — process — output
- use simple conditional statements
- use fixed loops

The National 5 learners could then be given additional tasks where they have to:

- use expressions to return values using arithmetic operations
- use selection constructs involving complex conditions
- use a mathematical pre-defined function

Data types and structures
For data types and structures, both National 4 and National 5 learners will cover string and numeric variables but National 5 learners might then be given additional exercises covering Boolean variables and 1-D arrays.

- **Design notations**

When National 4 and National 5 learners are designing their programs, both sets of learners will be taught the principles and importance of good design; National 4 learners might use a graphical notation whereas the National 5 learners would use pseudocode to exemplify programming constructs.

- **Testing and documenting solutions**

When it comes to testing and documenting solutions, both sets of learners will test their solutions, National 5 learners choosing their normal, extreme and exceptional test data but the National 4 learners will be given their test data. The National 5 learners will then document their programs using internal commentary.

In line with the underlying principles of Curriculum for Excellence, learners should be encouraged, and expected, to take an active role in their own learning. Where Course activities and materials allow them to progress in an independent manner, this will allow teaching of the two groups to happen most effectively.

Useful resources
Online resources (websites, microsites, wikis, newsfeeds, databases, etc) can provide a valuable source of easily accessible and up-to-date information on a wide range of software development tools and topics. In addition, the internet can act as a rich source of information for research into the impact of a software-based application on the environment or society.

Although not a definitive list, the following resources may support the delivery of the Software Design and Development (National 4) Unit.

Some suggested general online resources:

- Technology Student
- Teaching Education Scotland
- Education Scotland
- STEM Central on Education Scotland website
- Khan Academy

Some suggested specific online resources:

- Education Scotland Consolarium — there are a number of resources available here for games based learning, programming and software development activities
- Scratch — Scratch on the MIT website — within this site you can also search for Scratch projects by going to the featured projects area of the Channel section
- BYOB (Build Your Own Blocks) website
- Kodu Game Lab website
- Greenfoot website
- Serif website – DrawPlus tutorials
- Search in the app resources section to find specific tutorials
- MIT Centre for Mobile Learning website
- App Inventor software and tutorials
- Visual Basic 5 Control Creation Edition — search for tutorials on Google sites
- Dynamic Learning
- MonkeyJam PDF tutorial on the Brickshelf website — there are a number of tutorials available as pdfs within the gallery sections here
- other tutorials available on YouTube, including those for *MonkeyJam* and *TrackMania*
- How Stuff Works
- Raspberry Pi

Approaches to delivering and assessing each Outcome

The learner must demonstrate attainment of **both** of the Outcomes and their associated Assessment Standards. Assessment must be valid, reliable and fit for purpose.

SQA does not specify the methods of assessment to be used; teachers should determine the most appropriate method for their learners. In many cases, evidence (which may be oral or observational) will be gathered during normal classroom activities, rather than through formal assessment instruments.

Centres are expected to maintain a detailed record of evidence, including oral or observational evidence. Evidence in written or presentation format should be retained by the centre.

All evidence should be gathered under supervised conditions.

In order to ensure that the learner's work is their own, the following strategies are recommended:

- personal interviews with learners where teachers can ask additional questions about the completed work
- asking learners to do an oral presentation on their work
- ensuring learners are clear about acknowledging sources
- using checklists to record the authentication activity

Preparing learners for their assessment activities
In order to ensure that learners are prepared in advance for their assessment activities, it is good practice for teachers to make learners aware of the Assessment Standards required and to provide a range of feedback designed to improve learners' knowledge and skills as they progress through the Unit.

It is accepted as good practice that the evidence to meet the Assessment Standards for Outcome 1 and Outcome 2 be generated throughout the Course as an integral part of classroom activities.

Outcome 1

The learner will:

1 Explain how simple programs work, drawing on understanding of basic concepts in software development, by:

1.1 Reading and explaining code
1.2 Identifying the purpose of a range of programming constructs and how they work
1.3 Identifying the purpose of a range of variable types

The range of programming constructs should include expressions, sequence, selection and iteration.

Notes on delivery of Outcome 1
In order to meet Outcome 1, learners are expected to develop their knowledge and understanding of how programs work to the point where they have the ability to describe the purpose of the following constructs and to describe how they work:

- expressions to assign values to numeric and string variables
- sequence to determine the order in which statements are executed demonstrating input — process — output
- use of selection constructs, including simple conditional statements
- iteration using fixed loops

Learners should also develop the ability to read and interpret code by:

- identifying errors and suggesting corrections in a given piece of code
- interpreting and identifying the input, process and output in a given piece of code

Learners also need to be able to explain how data and instructions are stored:

- how assigning values to declared variables allows the value to be stored and used again within the program

Opportunities for different learning and teaching activities might include the following:

- Learners could be provided with working programs in a software development environment in order to be taught the purpose of a range of programming constructs, how they work and how data and instructions are stored.

- Teachers and lecturers could go over these constructs and how values are assigned to variables and then demonstrate how they work inside a program. This will develop the knowledge and understanding required in the learner to be able to read and explain sections of code within a program.

Notes on assessment of Outcome 1
Evidence of the Assessment Standards for Outcome 1 may be derived from a single, extended software development task, or from a number of shorter tasks, as an integral part of classroom activities.

- Learners can decide on the type of software programs to investigate.
- Learners can have access to books, the internet, pre-written code and other materials during the assessment.
- Learners should receive accurate and regular feedback from teachers and be actively involved in the assessment process.

Learners will use understanding of basic concepts in software development to explain how programs work. Written evidence may take the form of responses to a series of assignments or a short test in which the learner demonstrates the ability to:

- read and interpret code
- identify the purpose of a range of programming constructs
- identify the purpose of a range of variable types

A valid and reliable form of assessment would, for example, present the learner with a section of code which contains, e.g., examples of selection and the use of string and numeric variables. It would then require the learner to:

- identify the purpose of the selection constructs in the coding
- describe how those constructs work
- identify the purpose of the string and numeric variables

The learner would be expected to produce an accurate series of descriptions in the form of extended responses.

Note that the coding presented to the learner should include a representative sample of the programming constructs and variable types set out in the mandatory content.

Evidence of the Assessment Standards for Outcome 1 may be oral or written. Evidence in written form should be retained by the centre.

Where learners' responses have been in oral form, centres are expected to keep either a recording of learners' performance as evidence and/or an observation checklist.

All evidence should be gathered in supervised conditions.

Outcome 2

The learner will:

2 Develop short programs using a software development environment by:

2.1 Selecting and using expressions, sequence, selection and iteration
2.2 Selecting and using appropriate simple data types, such as numeric (integer) and string
2.3 Testing digital solutions using supplied test data

Programs should include at least one construct and one data type.

Notes on delivery of Outcome 2
To meet Outcome 2, learners are expected to gain experience of using different software development environments, although they may produce evidence from only one if they so wish.

The advantage of using more than one development environment is that it enables comparison between (say) the operation of constructs in two or more environments, providing opportunities to deepen the learner's understanding. Using just one development environment, however, might secure deeper development of knowledge and skills.

Some software development environments available have a more extensive range of programming features than others. A software development environment with a narrowly restricted range of features would not be suitable for meeting the demands of this Unit.

Below is a non-restrictive list of possible examples of software development environments which might be suitable. The choice of software development environment is entirely at the discretion of the centre and should be based on the suitability of the chosen environment to support the delivery of the mandatory content of the Unit.

Non-restrictive examples of current software development environments
Graphical environments in which code is assembled by combining graphical objects which represent instructions, variables and constructs, such as Scratch
Software development environments which are specifically suited to games development, such as C++, Gamemaker, Greenfoot and Dark Basic
Apps development environments which are suited to producing applications for handheld devices and smartphones, such as Java development tools, App Inventor
Text-based programming environments, such as Visual Basic and Truebasic

Developing short programs
Learners could be asked to develop short programs using the following constructs in a software development environment:

- expressions to assign values to variables
- execution of lines of code in sequence demonstrating input — process — output
- use of selection constructs including simple conditional statements
- iteration and repetition using fixed loops

Level of complexity of programs
In this National 4 Unit it is expected that learners' programs should contain at least one data type, either string or numeric where appropriate, and at least one construct.

Testing digital solutions
Learners should develop skills in testing their programs and learn to use supplied sets of test data. They should:

- test their programs using test data which is supplied by the teacher
- identify and rectify errors in programs

Notes on assessment of Outcome 2
Evidence is only required of one successful example for each Assessment Standard and can be generated throughout the Unit as an integral part of classroom activities, using a single problem-solving task or by a series of separate activities.

Learners will develop a number of short programs in a software development environment and evidence can be gathered from any of these throughout the duration of the Unit.

- Learners can decide which type of software development environment to work in.
- Learners can have access to books, the internet and other materials during the assessment.
- Learners should receive accurate and regular feedback from teachers and lecturers and be actively involved in the assessment process.

The short programs should be completed and each should include both of the following:

- at least one construct
- one data type

Teachers and lecturers will select the appropriate evidence from the learner on the following aspects:

- The solutions to the tasks set should be straightforward and, for example, not involve complex combinations of selection and iteration.
- Any conditions which are part of selection or iteration constructs are expected to be simple in structure and not to require the use of logical operators.

Learners should:

- use supplied test data
- identify and correct errors

Users should produce their programs in supervised conditions.

Evidence of the Assessment Standards for Outcome 2 may be derived from several software development tasks; formal documentation is not expected or required. However, an observation checklist should be maintained by the teacher to record learners' performance.

While formal documentation is not required, it is recognised good practice to keep digital copies of all learners' work in order to support the learning process by acting as a source of reference and revision activities, particularly if the Unit is being taken as part of the National 4 Computing Science Course.

Added value

Courses from National 4 to Advanced Higher include assessment of added value. At National 4 the added value will be assessed in the Added Value Unit.

Information given in the *Added Value Unit Specification* about the assessment of added value is mandatory.

Ii this Unit is being taken as part of the National 4 Computing Science Course then it is important to emphasise that all of the mandatory content specified in the Software Design and Development table in the 'Further information on Course coverage' section of the *Added Value Unit Specification* should be covered in the course of this Unit.

Developing skills for learning, skills for life and skills for work

Learners are expected to develop broad generic skills as an integral part of their learning experience. The Unit Specification lists the skills for learning, skills for life and skills for work that learners should develop through this Course. These are based on SQA's Skills Framework: Skills for Learning, Skills for Life and Skills for Work and must be built into the Unit where there are appropriate opportunities. The level of these skills will be appropriate to the level of the Unit.

The table below highlights opportunities to develop these skills during this Unit.

2 Numeracy	
2.1 Number processing	Learners can be given opportunities to develop their number processing skills by gaining practice in problem solving in numeric-based contexts which involve eg multiplication, division or calculating percentages. Problem-solving contexts could then be set in which software would take decisions and vary the output based on the results of calculations.
2.3 Information handling	Information handling skills could be developed by setting problem-solving contexts in which learners are required to use data set out in tables or a graphical format as the basis for input to their programs which then process the data to produce required output.

4 Employability, enterprise and citizenship	
4.2 Information and communication technology	Throughout the Unit learners will be continuously interacting with the technology around them and will be given abundant opportunities to extend their ICT skills.

5 Thinking skills	
5.3 Applying	Learners will be given opportunities to apply their software-based knowledge and skills and then test their solutions.

Combining assessment within Units

Holistic assessments could be developed which cover some of the Assessment Standards from Outcomes 1 and 2.

For example: an assessment could be devised that asks the learner to create a program with at least one construct, at least one data type and some variables. Once they have created the program, tested it, and rectified any errors, they could then be asked to describe the purpose of the constructs and how they work. The assessor could then select a segment of code from the program and ask the learner to explain what the code actually does. The assessor could also ask the learner to explain how data and instructions are stored.

This would cover the following Assessment Standards:

1.1 Reading and explaining code
1.2 Identifying the purpose of a range of programming constructs and how they work
1.3 Identifying the purpose of a range of variable types

2.1 Selecting and using expressions, sequence, selection and iteration
2.2 Selecting and using appropriate simple data types such as numeric (integer) and string
2.3 Testing digital solutions using supplied test data

Equality and inclusion

The requirement to develop practical skills involving the use of equipment and tools may present challenges for learners with physical or visual impairment. In such cases, reasonable adjustments may be appropriate, including (for example) the use of adapted equipment or alternative assistive technologies. This is for both candidates and centres to consider.

It is recognised that centres have their own duties under equality and other legislation and policy initiatives. The guidance given in these *Unit Support Notes* is designed to sit alongside these duties but is specific to the delivery and assessment of the Unit.

Alternative approaches to Unit assessment to take account of the specific needs of learners can be used. However, the centre must be satisfied that the integrity of the assessment is maintained and that the alternative approaches to assessment will, in fact, generate the necessary evidence of achievement.

Centres will find more guidance on this in the series of publications on Assessment Arrangements on SQA's website: www.sqa.org.uk/sqa//14977.html.

Appendix 1: Reference documents

The following reference documents will provide useful information and background.

- Assessment Arrangements (for disabled candidates and/or those with additional support needs) — various publications on SQA's website: http://www.sqa.org.uk/sqa/14976.html
- Building the Curriculum 4: Skills for learning, skills for life and skills for work
- Building the Curriculum 5: A framework for assessment
- Course Specifications
- Design Principles for National Courses
- Guide to Assessment (June 2008)
- Overview of Qualification Reports
- *Overview of Qualification Reports*
- *Principles and practice papers for curriculum areas*
- *Research Report 4 — Less is More: Good Practice in Reducing Assessment Time*
- *Coursework Authenticity — a Guide for Teachers and Lecturers*
- SCQF Handbook: User Guide *(published 2009)* and SCQF level descriptors (to be reviewed during 2011 to 2012): www.sqa.org.uk/sqa/4595.html
- SQA Skills Framework: Skills for Learning, Skills for Life and Skills for Work
- Skills for Learning, Skills for Life and Skills for Work: Using the Curriculum Tool
- SQA Guidelines on e-assessment for Schools
- SQA Guidelines on Online Assessment for Further Education
- SQA e-assessment web page: www.sqa.org.uk/sqa/5606.html

Administrative information

Published: May 2016 (version 2.0)

Superclass: CB

History of changes to Unit Support Notes

Version	Description of change	Authorised by	Date
1.1	Clarifications to mandatory content in Course Support Notes; minor changes to *Software Design and Development* Unit Support Notes; addition of 'storage' to Assessment Standard 2.2 in *Information System Design and Development* Unit Support Notes.	Qualifications Development Manager	July 2013
2.0	Changes made throughout the document to reflect revisions made to Assessment Standards in Outcomes 1 and 2, and removal of Outcome 3 and Assessment Standard 2.4 'Identifying and rectifying errors in programs' in Outcome 2.	Qualifications Manager	June 2015

Unit Support Notes

Unit Support Notes — Information System Design and Development (National 4)

Please refer to the note of changes at the end of this document for details of changes from previous version (where applicable).

Introduction

These support notes are not mandatory. They provide advice and guidance on approaches to delivering and assessing the *Information System Design and Development* (National 4) Unit. They are intended for teachers and lecturers who are delivering this Unit. They should be read in conjunction with:

- the Unit Specification
- the Course Specification
- the Course Assessment Specification
- the Course Support Notes
- appropriate assessment support materials

General guidance on the Unit

Aims

The purpose of this Unit, as stated in the *Unit Specification*, is for the learner to develop basic knowledge, understanding and practical problem-solving skills related to the design and development of information systems through a range of practical and investigative tasks. Learners will implement practical solutions using a range of development tools. They will also develop an understanding of hardware, software, connectivity and security issues, and relate these to one or more information systems.

This Unit will give learners the opportunity to develop their thinking skills as well as their skills in numeracy, employability, enterprise and citizenship.

This Unit can be delivered:

- as a stand-alone Unit
- as part of the National 4 Computing Science Course

Progression into this Unit

Entry to this Unit is at the discretion of the centre. However, learners would normally be expected to have attained some useful skills and knowledge from prior learning, such as:

- National 3 Computing Science Course or relevant component Units
- National 3 Numeracy

Learners may also have gained relevant skills and knowledge through other education systems or from their own interests and informal learning.

Skills, knowledge and understanding covered in this Unit

Information about skills, knowledge and understanding is given in the National 4 Computing Science *Course Support Notes.*

If the Unit is being delivered as part of the National 4 Computing Science Course, the teacher should refer to the 'Further mandatory information on Course coverage' section within the *Added Value Unit Specification* for detailed content.

If this Unit is being delivered on a free-standing basis, teachers are free to select the skills, knowledge, understanding and contexts which are most appropriate for delivery in their centres.

Progression from this Unit

On successful completion of this Unit, the following Units and Courses provide appropriate progression pathways for learners:

- Information System Design and Development (National 5) Unit
- National 5 Computing Science Course
- National Certificate Group Awards in a range of Computing, IT and related disciplines
- National Progression Awards in Digital Media
- other technological Courses at National 4
- employment, apprenticeships and/or training in Computing, IT and related fields

Approaches to learning, teaching and assessment

Learning and teaching
The Unit is designed to provide flexibility, personalisation and choice for both the learner and the teacher.

Learning and teaching activities should be designed to stimulate learners' interest, and to develop skills and knowledge to the standard required by the Outcomes and to the level defined by the associated Assessment Standards. Learning should be supported by appropriate practical activities so that skills are developed simultaneously with knowledge and understanding.

The *Unit Specification* defines the skills and knowledge required, but leaves complete freedom to the teacher and learner to select interesting contexts and environments in which to develop these.

An investigatory approach is encouraged, with learners actively involved in developing their skills, knowledge and understanding by investigating a range of real-life and relevant information systems, problems and solutions.

Practical activities and investigations lend themselves readily to group work, and this should be encouraged. Individual, paired or group problem-solving tasks could be related to authentic and relevant contexts. For example, learners could be asked to:

- create a database structure to hold information such as a CD or MP3 collection
- create a website that appeals to the viewer with appropriate use of text, graphics, sound and video similar to the glossy magazine websites
- work in pairs or groups to research the different types of user interface for information systems aimed at different users
- work in pairs or groups to compare the security risks of particular types of information system, eg office information systems, decision support systems, transaction processing systems, management reporting systems

The National 4 Computing Science *Course Support Notes* provide further broad guidance on approaches to learning and teaching which may apply to both of the Units of the Course and should be read before delivering this Unit.

When delivering the Unit as part of the National 4 Computing Science Course, reference should be made to the appropriate content statements within the 'Further mandatory information on Course coverage' section of the *Added Value Unit Specification* to ensure the required breadth of knowledge is covered

Sequence of delivery of Outcomes
The sequence of delivery of the Outcomes is a matter of professional judgement and is entirely at the discretion of the centre. Some suggested approaches are listed below.

Outcome 1 and Outcome 2 simultaneously
Teachers may consider it good practice to combine the development of knowledge and understanding of the factors involved in the design and

implementation of an information system at the same time as learners progress through the practical tasks involved in developing simple information systems.

For example, when learners are designing their information systems, whether databases, websites, or other types of information systems, they will have to consider the functionality and features of the products they are designing. Learners may experiment with different types of software applications to create their products and they should be made aware of the hardware and software requirements for these different types of applications. Although learners will probably use software that is available to them on their centre's network, they should also have an idea of the type of similar applications that are available from cloud systems. File sizes and types of storage media should be considered when learners are storing their products on the network, which highlights the importance of storage and capacity for any information system.

Outcome 1 before Outcome 2
It is possible to develop the learner's practical skills in creating information systems and meet the requirements of Outcome 1 before moving to formally develop the knowledge and understanding associated with the factors involved in the design and implementation of an information system in Outcome 2.

For example, learners could develop the practical skills associated with creating information systems by working with databases, websites, or other information systems. This would allow them to become familiar with the features of different software applications used to build these systems; how to integrate different media types and how to identify and rectify errors and bugs that occur during implementation. Following this process could give them a solid understanding of the main factors involved in developing a 'real-life' information system and the practical skills required to actually create it. By investigating the technical specification and the security risks of a 'real-life' information system, learners should be able to relate the factors and concepts to the smaller-scale information systems that they have created.

Learners should be encouraged to make connections between the processes they went through in developing their own information systems and the factors that have to be considered in the design and implementation of information systems used in the real world.

Outcome 2 before Outcome 1
It is also possible to consider the factors involved in the design and implementation of an information system and meet the requirements of Outcome 2 before actually carrying out the practical task of creating information systems.

For example, learners could develop an understanding of the basic factors involved in the design and implementation of a 'real-life' information system before developing their own. They could select a particular information system that is used in their school, a local company or business and then find out the following:

- What are its main features?
- What functionality does it offer?
- What are the hardware, software, storage and connectivity requirements?
- What are the main security risks with the system?

With this overview in mind, learners could then proceed to create their own simple information systems, such as databases, websites, multimedia applications and/or a hybrid of these.

By doing Outcome 2 first, learners should be able to relate the basic issues and factors they discovered when researching a 'real-life' information system to their own products, albeit on a much smaller scale. Learners could then develop their practical skills by discovering the functionality of the software they are using to create structures with links and integrating different media types. They might look at the issues related to the importance of storage requirements in real-life information systems.

Meeting the needs of all learners
When teaching this Unit to a class, with some learners working towards National 4 and others towards National 5, it may be useful for the teacher to identify activities covering common learning activities all learners, and additional activities for National 5 learners. For example:

- **Structures and links**

Both National 4 and National 5 learners will be creating information systems, such as databases, websites, or other types of information systems, but the National 5 learners will have different specifications. Both sets of learners can decide on the type of information systems they are going to create and the types of software they are going to use to build the systems. Teacher-directed lessons or self-supported materials could be used to guide the learner through the main functions of selected software with extension exercises built-in for the National 5 learners dealing with such aspects as linked tables; link and Boolean data types; field lengths and range checks.

- **User interface**

Lessons covering common concepts such as the importance of applying good design could be delivered to both sets of learners, with the National 5 learners studying additional criteria for good user interface, such as visual layout, navigation, selection, consistency, interactivity and readability.

- **Media types**

Both sets of learners will need to integrate text, graphics, sound and video into their information systems, but National 5 learners should be made aware of the properties of standard file formats, factors affecting file size and quality, and the need for compression.

- **Coding**

National 4 learners do not need to code, but at National 5 learners should learn the basic constructs of a scripting language such as JavaScript and a mark-up language such as HTML.

- **Technical implementation/security**

National 4 and National 5 learners should both study the technical specifications of real-life information systems covering the same aspects in terms of processor type, speed, memory (RAM), input and output devices; however, the National 5 learners are also required to consider security precautions, legal aspects and impact on the environment

- **Technical implementation/networking and connectivity**

On the topic of networking and connectivity, National 4 learners might compare stand-alone computers with networked computers, LAN with the internet and wired with wireless connections. National 5 learners should also be comparing networks but they might be looking at peer-to-peer and client/server set-ups as well.

Different contexts

Where National 5 learners have studied National 4 in a previous year, it is important to provide them with new and different contexts for learning to avoid de-motivation. It is particularly important that learners do not feel that they are simply doing the same work over again, albeit at a deeper level.

For example, at National 4, learners will have built simple information systems, such as databases, websites, or other types of information systems. At National 5, they will be able to add more functionality to one or more products or build new systems using the techniques and skills they have acquired throughout the National 5 Course. They will be able to include internal and external hyperlinks and apply good navigational design. Learners may have focused mainly on (say) web applications at National 4, and then could spend more time studying (say) database systems at National 5.

Computational thinking

Aspects of computational thinking — abstraction, algorithms, decomposition, pattern recognition and generalisation — are relevant to the design and development of information systems, just as they are in programming, and so should be exemplified within this Unit where appropriate.

In line with the underlying principles of Curriculum for Excellence, learners should be encouraged, and expected, to take an active role in their own learning. Where Course activities and materials allow them to progress in an independent manner, this will allow teaching of the two groups to happen most effectively.

Useful resources

Online resources (websites, microsites, wikis, newsfeeds, databases, etc) can provide a valuable source of easily accessible and up-to-date information on a wide range of information systems design and develop topics. In addition, the internet can act as a rich source of information for research into the security risks involved in digital communication.

Although not a definitive list, the following resources may support the delivery of the Information System Design and Development (National 4) Unit.

Some suggested general online resources:

- Technology Student
- Teaching Education Scotland
- Education Scotland
- STEM Central on Education Scotland website
- Khan Academy

Some suggested specific online resources:

- Education Scotland Consolarium — there are a number of resources available here for games based learning, programming and software development activities
- Scratch — Scratch on the MIT website — within this site you can also search for Scratch projects by going to the featured projects area of the Channel section
- BYOB (Build Your Own Blocks) website
- Kodu Game Lab website
- Greenfoot website
- Java website
- Serif website — DrawPlus tutorials
- Search in the app resources section to find specific tutorials
- MIT Centre for Mobile Learning website
- App Inventor software and tutorials
- Visual Basic 5 Control Creation Edition – search for tutorials on Google sites
- Dynamic Learning
- MonkeyJam PDF tutorial on the Brickshelf website — there are a number of tutorials available as pdfs within the gallery sections here
- other tutorials available on YouTube, including those for *MonkeyJam* and *TrackMania*
- Hackasaurus website
- How Stuff Works
- Raspberry Pi

Approaches to delivering and assessing each Outcome

The learner must demonstrate attainment of all of the Outcomes and their associated Assessment Standards. Assessment must be valid, reliable and fit for purpose.

SQA does not specify the methods of assessment to be used; teachers should determine the most appropriate method for their learners. In many cases, evidence (which may be oral or observational) will be gathered during normal classroom activities, rather than through formal assessment instruments.

Centres are expected to maintain a detailed record of evidence, including oral or observational evidence. Evidence in written or presentation format should be retained by the centre.

All evidence should be gathered under supervised conditions.

In order to ensure that the learner's work is their own, the following strategies are recommended:

- personal interviews with learners where teachers can ask additional questions about the completed work
- asking learners to do an oral presentation on their work
- ensuring learners are clear about acknowledging sources
- using checklists to record the authentication activity

Preparing learners for their assessment activities
In order to ensure that learners are prepared in advance for their assessment activities, it is good practice for teachers to make learners aware of the Assessment Standards required and to provide a range of feedback designed to improve learners' knowledge and skills as they progress through the Unit.

It is accepted as good practice that the evidence to meet the Assessment Standards for Outcome 1 and Outcome 2 be generated throughout the Course as an integral part of classroom activities.

Outcome 1

The learner will:

1 Develop simple information systems, using appropriate development tools by:

1.1 Creating a structure and links
1.2 Integrating different media types

Notes on delivery of Outcome 1
It is envisaged that learners will develop a number of different information systems, which might include databases, web pages, multimedia products, web-based applications or other current information systems. Developing a range of these would allow the learner to develop the broad knowledge and understanding required for Course assessment. Tasks and activities throughout the Unit should be linked to relevant contexts.

Learners could work in pairs or groups to consider and discuss the basic features and functionality of a good information system from both the operator's and end-user's perspectives.

Learners should be given instruction or resources to enable them to use software applications to incorporate and integrate different media types into an information system.

Practical tutorials or teacher-led instruction may be required to teach the basics, followed by a series of tasks or exercises to reinforce learners' understanding. Learners should be encouraged to identify any errors and seek solutions during the development of their information systems.

Notes on assessment of Outcome 1
Evidence is only required of one successful example for each Assessment Standard and can be generated throughout the Unit as an integral part of classroom activities, using a single problem-solving task or by a series of separate activities.

- Learners can decide which type of information systems to develop, eg databases, a series of web pages, multimedia products, web-based applications or other current information systems.
- Learners can have access to books, the internet and other materials during assessment activities.

- Learners should receive accurate and regular feedback from teachers and lecturers and be actively involved in the assessment process.

Teachers will select the appropriate evidence from the learner. Evidence can be gathered from observation; however, centres should keep a record, either in digital format or as an observational checklist, recording, video, etc for verifying purposes.

All evidence should be gathered under supervised conditions.

Outcome 2

The learner will:

2 Consider a number of basic factors involved in the design and implementation of an information system by:

2.1 Describing its basic features and functionality
2.2 Describing its hardware, software, storage and connectivity requirements
2.3 Describing relevant security risks

Notes on delivery of Outcome 2
It is envisaged that learners will investigate a real-life information system (this might be an office information system, a decision support system, a transaction processing system, management reporting system, a website or other), and by focusing on the basic factors that should be considered in the design and implementation of this information system they will meet the Assessment Standards for Outcome 2.

Teachers can provide some background information relating to aspects of the information system to be investigated. Learners have to demonstrate that they have a detailed understanding of the underlying concepts of this information system, including hardware and software, storage and connectivity requirements and the security risks involved.

Opportunities for different learning and teaching activities might include:

- comparing different devices, such as desktops, laptops, tablet PCs and smartphones in terms of processor speed and memory (RAM, ROM
- using online research skills to explore websites such as Dell, PC World, etc to compare the range of different storage devices available
- researching the news online to look for instances when security risks have occurred through digital communication modes

Outcome 2 lends itself to more pair and/or group work. Aspects of existing information systems solutions to real-world problems can be analysed in simple terms to aid understanding, and a number of different types of 'real-life' information systems should be provided to the learners. The tasks and activities here will be more of the research and investigation type, with learners working in pairs or groups.

For example, each group could be asked to analyse the features of the information system in their school or college to obtain an overview of the many tasks that it performs.

Alternatively, each group could be given an information system to investigate and report on, under the following headings:

- features and functionality
- hardware, software, storage and connectivity
- security risks

Notes on assessment of Outcome 2
Evidence for Outcome 2 can be generated throughout the Unit as an integral part of classroom activities.

- Learners can decide on the information system to investigate.
- Learners can have access to books, the internet and other materials during the assessment.
- Learners should receive accurate and regular feedback from teachers and lecturers and be actively involved in the assessment process.

The learner would be expected to produce an accurate series of descriptions in the form of extended responses. However, the evidence for Outcome 2 need not be written, but may be presented in another format, such as a text document, a series of web pages, a visual presentation, a video, a podcast, a blog or any other appropriate format.

Teachers and lecturers will select the appropriate evidence from the learner on the following aspects of the information system:

2.1 statement relating to features and functionality of the information system
2.2 description of the hardware, software, storage and connectivity required to run the information system
2.3 list of the security risks involved in accessing information systems to the operator or end-users

Where a learner's responses have been in oral form, centres should keep either a recording of a learner's performance as evidence and/or an observation checklist.

All evidence should be gathered in supervised conditions.

Added value

Courses from National 4 to Advanced Higher include assessment of added value. At National 4 the added value will be assessed in the Added Value Unit.

Information given in the *Added Value Unit Specification* about the assessment of added value is mandatory.

If this Unit is being taken as part of the National 4 Computing Science Course, it is important to emphasise that all of the mandatory content specified in the Information System Design and Development table in the 'Further information on Course coverage' section of the *Added Value Unit Specification* should be covered in the course of this Unit.

Developing skills for learning, skills for life and skills for work

Learners are expected to develop broad generic skills as an integral part of their learning experience. The *Unit Specification* lists the skills for learning, skills for life and skills for work that learners should develop through this Course. These are based on SQA's *Skills Framework: Skills for Learning, Skills for Life and Skills for Work* and must be built into the Unit where there are appropriate opportunities. The level of these skills will be appropriate to the level of the Unit.

The table below highlights opportunities to develop these skills during this Unit.

2 Numeracy	
2.1 Number processing	Learners can be given opportunities to develop their number processing skills by gaining practice in problem solving in numeric-based contexts, eg calculation of main memory, storage requirements, and then making informed decisions based on the results of these calculations.
2.3 Information handling	Information handling skills could be developed by setting problem-solving contexts in which learners are required to interpret data in different structures, for example, fields, records and files in databases, URLs and hyperlinks in web pages, and comparison of current input/output devices using the criteria of speed and memory.

4 Employability, enterprise and citizenship	
4.2 Information and communication technology	Throughout the Unit, learners will be continuously interacting with the technology around them and will be given abundant opportunities to extend their ICT skills. When investigating an information system for Outcome 2, they will be given opportunities to specifically develop their ICT-based research skills.

5 Thinking skills	
5.3 Applying	Learners will be given ample opportunities to analyse a wide range of problems, apply the knowledge and skills they have acquired and then test their solutions.

Combining assessment within Units

It may be possible to develop learning/assessment activities which provide evidence that learners have achieved the standards for more than one Outcome within the Unit, thereby reducing the assessment burden on learners. Combining assessment of Outcomes (or parts of Outcomes) in this way is perfectly acceptable, but needs to be carefully managed to ensure that all Assessment Standards and Outcomes for the Unit are covered.

Equality and inclusion

The requirement to develop practical skills involving the use of equipment and tools may present challenges for learners with physical or visual impairment. In such cases, reasonable adjustments may be appropriate, including (for example) the use of adapted equipment or alternative assistive technologies. This is for both candidates and centres to consider.

It is recognised that centres have their own duties under equality and other legislation and policy initiatives. The guidance given in these *Unit Support Notes* is designed to sit alongside these duties but is specific to the delivery and assessment of the Unit.

Alternative approaches to Unit assessment to take account of the specific needs of learners can be used. However, the centre must be satisfied that the integrity of the assessment is maintained and that the alternative approaches to assessment will, in fact, generate the necessary evidence of achievement.

Centres will find more guidance on this in the series of publications on Assessment Arrangements on SQA's website: www.sqa.org.uk/sqa/14977.html.

Appendix 1: Reference documents

The following reference documents will provide useful information and background.

- Assessment Arrangements (for disabled candidates and/or those with additional support needs) — various publications on SQA's website: http://www.sqa.org.uk/sqa/14976.html
- British Computer Society, *Glossary of Computing and IT*, 12th edition
- Building the Curriculum 4: Skills for learning, skills for life and skills for work
- Building the Curriculum 5: A framework for assessment
- Course Specifications
- Design Principles for National Courses
- Guide to Assessment (June 2008)
- Overview of Qualification Reports
- *Overview of Qualification Reports*
- *Principles and practice papers for curriculum areas*
- *Research Report 4 — Less is More: Good Practice in Reducing Assessment Time*
- *Coursework Authenticity — a Guide for Teachers and Lecturers*
- SCQF Handbook: User Guide *(published 2009)* and SCQF level descriptors (to be reviewed during 2011 to 2012): www.sqa.org.uk/sqa/4595.html
- SQA Skills Framework: Skills for Learning, Skills for Life and Skills for Work
- Skills for Learning, Skills for Life and Skills for Work: Using the Curriculum Tool
- SQA Guidelines on e-assessment for Schools
- SQA Guidelines on Online Assessment for Further Education
- SQA e-assessment web page: www.sqa.org.uk/sqa/5606.html

Administrative information

Published: May 2016 (version 2.1)

Superclass: CB

History of changes to Unit Support Notes

Version	Description of change	Authorised by	Date
1.1	Clarifications to mandatory content in Course Support Notes; minor changes to *Software Design and Development* Unit Support Notes; addition of 'storage' to Assessment Standard 2.2 in *Information System Design and Development* Unit Support Notes.	Qualifications Development Manager	July 2013
2.0	Removal of Assessment Standard 1.3 in Outcome 1 and the revision of all Assessment Standards in Outcome 2.	Qualifications Manager	June 2015
2.1	For the ISDD Unit, 'input and output' removed from 'Notes on delivery of Outcome 2'.	Qualifications Manager	May 2016