Computing Science progression framework

This document provides advice and guidance on how experiences and outcomes at third and fourth curriculum levels relate to the Outcomes and Assessment Standards of National 4 and National 5 qualifications. It also provides advice and guidance on how key concepts in Computing Science, including computational thinking*, can be developed as learners progress through curriculum and qualification levels.

Suggested topics and/or activities are provided to give a broad indication of depth of treatment at each level and the nature of progression between levels. They are not intended to outline a program of learning and teaching and do not attempt to describe progression within a level.

This document is intended as advice and should not be seen as being prescriptive in any way.

Mandatory requirements for assessment are detailed in Course Specifications, Unit Specifications, Added Value Unit Specifications and Course Assessment Specifications.

For the purposes of this document, experiences and outcomes, and Outcomes and Assessment Standards, have been grouped under five main headings:

- Designing and creating computer programs
- Information storage and communication
- Computer systems and architecture
- Security considerations
- Impact of technological developments on society and environment

Other systems of grouping are possible. However, these headings match areas of study that can be found in many learning programmes.

*There are various ways of defining computational thinking. However, the following inter-related aspects are seen as being important:

- **Abstraction:** seeing a problem and its solution at many levels of detail.
- **Algorithmic thinking:** the ability to develop a step-by-step strategy for solving a problem.
- **Decomposition:** breaking down a problem into sub-problems.
- **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 is the idea that computers are **deterministic**: it is possible to predict what they will do.

Designing and creating computer programs				
Related experiences, outcomes and Assessment Standards	**Third level experiences and outcomes**	**Fourth level experiences and outcomes**	**National 4 Outcomes and Assessment Standards**	**National 5 and Outcomes and Assessment Standards**
	Using appropriate software, I can work individually or collaboratively to design and implement a game, animation or other application. (TCH 3-09a)	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)	**(SDD O1.1, O1.2)** **Explain how simple programs work, drawing on understanding of basic concepts in software development by:** ♦ Reading and explaining code ♦ Describing the purpose of a range of programming constructs and how they work **(SDD O2)** **Develop short programs using a software development environment by:** ♦ Selecting and using expressions, sequence, selection and iteration ♦ Selecting and using appropriate simple data types, such as numeric (integer) and string ♦ Testing digital solutions using supplied test data ♦ Identifying and rectifying errors in programs **(SDD O3.1, O3.2)** **Produce a short factual report on a contemporary software-based application by:** ♦ Describing the application ♦ Explaining how its features relate to programming constructs and data types	**(SDD O1.1, O1.2, O1.3)** **Explain how programs work, drawing on understanding of concepts in software development and basic computer architecture by:** ♦ Reading and explaining code ♦ Describing the purpose of a range of programming constructs and how they work ♦ Describing the purpose and role of variables **(SDD O2)** **Develop short programs using one or more software development environments by:** ♦ Selecting and using a combination of expressions, sequence, selection, iteration and pre-defined functions ♦ Selecting and using appropriate simple data types, such as numeric (integer and real), string and Boolean ♦ Testing digital solutions using own test data ♦ Identifying and rectifying errors in programs ♦ Providing internal commentary or documentation **(SDD O3.1, O3.2, O3.3)** **Produce a short detailed report comparing two contemporary software development languages or environments by:** ♦ Describing how each represents standard constructs ♦ Comparing the range of data types provided ♦ Comparing their editing features
Possible topics/activities	**Producing a solution** Using video editing software to produce a stop-motion animation.	**Understanding code** Given a simple problem and sample code, identify errors: ♦ Incorrect or missing keywords ♦ Inconsistency in variable or procedure naming	**Understanding code** Given several simple programs, identify and explain the operation of each. Programs may make use of the following constructs and data types/structures: *Constructs* ♦ Conditional loop ♦ Fixed loop ♦ If…then…else using simple conditions ♦ Assignment statements and arithmetic operations *Data types/structures* ♦ string ♦ integer	**Understanding code** Given several simple programs, identify and explain the operation of each program but adding internal commentary to the code. Programs may make use of the following constructs and data types/structures: *Constructs* ♦ Conditional loop ♦ Fixed loop ♦ If…then…else using simple and complex conditions ♦ Assignment statements and arithmetic operations ♦ Pre-defined functions

	Developing code Write a simple game, with support, using a visual/blocks-based programming language, eg Scratch, Kodu Game Lab or similar.	**Developing code** Write a simple game or other application, making use of own images and/or animations. Use of a suitable software development environment, eg Stencyl, Greenfoot or similar. **Evaluating and testing code** Evaluate the game or application created by producing a list of aspects to improve.	Given several simple programs, identify and rectify errors such as: ♦ Incorrect or missing keywords ♦ Inconsistency in variable or procedure naming ♦ Invalid expressions **Developing code** Given a partially complete short program and its specification, write the code to complete the program. **Evaluating and testing code** Test completed program using a set of supplied test data.	*Data types/structures* ♦ Boolean variables ♦ 1-D arrays ♦ string ♦ integer, real Given several simple programs, identify and rectify errors such as: ♦ Incorrect or missing keywords ♦ Inconsistency in variable or procedure naming ♦ Invalid expressions ♦ Syntax errors **Developing code** Given a specification, develop a short program which uses 1D arrays to store and manipulate data. Program should also carry out input validation. **Evaluating and testing code** Test program using user-generated normal, extreme and exceptional test data. **Development environments** Development environments to consider: Greenfoot, Stencyl, Eclipse, Xcode, MIT App Inventor, Scratch/BYOB, Dreamweaver and others. Languages to consider: Java, JavaScript, PHP, Python and others.
Aspects of computational thinking: abstraction	Objects and operations provided by development/authoring software to create game/animation/application. Design of game/animation/application in terms of these objects and operations.	Computational constructs provided by language, for example to represent and store data items, and control program flow. Design of programs in terms of computational constructs. [Objects and operations provided by development software to create animations.]	Computational constructs provided by language, for example to represent and store data items, and control program flow. Design of programs in terms of computational constructs. Using design tools to translate straightforward problem statements into more abstract representations, and ultimately to code.	Computational constructs provided by language, for example to represent and store data items, control program flow and encapsulate sub programs. Design of programs in terms of computational constructs. Using design tools to translate increasingly complex problem statements into more abstract representations, and ultimately to code.
Aspects of computational thinking: algorithmic thinking	Using simple programming languages/environments to create games, animations or other applications. [Using development/authoring software to create animations.]	Creation of simple algorithms in the design and creation of simple programs. Sequence, selection and repetition. [Sequence of frames to create animation.]	Creation of simple algorithms in the design and creation of simple programs. Sequence, selection and repetition.	Creation of algorithms in the design and creation of programs. Sequence, selection and repetition. Modularity (pre-defined functions).
Aspects of computational thinking: decomposition			Breaking problem down into sub-problems to develop algorithm.	Breaking problem down into sub-problems.
Aspects of computational thinking: pattern recognition			Identifying how familiar constructs can be used to solve similar problems.	Identifying how familiar constructs can be used to solve similar problems.
Aspects of computational thinking: generalisation		Use of simple data types (variables) to make programs re-usable.	Use of simple data types (variables) to make programs re-usable.	Use of parameters in pre-defined functions.
Deterministic nature of computer	Output/behaviour can be predicted.	Predicting output. Testing. Rectifying errors.	Predicting output. Testing. Rectifying errors. Explaining program behaviour.	Predicting output. Testing. Rectifying errors. Explaining program behaviour.

Information storage and communication				
Related experiences, outcomes and Assessment Standards	**Third level experiences and outcomes**	**Fourth level experiences and outcomes**	**National 4 Outcomes and Assessment Standards**	**National 5 and Outcomes and Assessment Standards**
	I can build a digital solution which includes some aspects of multimedia to communicate information to others. (TCH 3-08b)	I can integrate different media to create a digital solution which allows interaction and collaboration with others. (TCH 4-08c) I can explore the properties and functionality of [software] to establish their suitability for a task at home or in the world of work. (TCH 4-14c)	**(ISDD O1)** **Develop simple information systems, using appropriate development tools, by:** ♦ Creating a structure and links ♦ Integrating different media types ♦ Identifying and rectifying errors **(ISDD O2.1)** **Consider a number of basic factors involved in the design and implementation of an information system by describing, in simple terms:** ♦ its basic features and functionality	**(ISDD O1)** **Develop information systems, using appropriate development tools by:** ♦ Creating a structure and links ♦ Creating a user interface ♦ Writing or editing simple code ♦ Integrating different media types ♦ Identifying and rectifying errors **(ISDD O2.1)** **Consider the factors involved in the design and implementation of an information system by describing its:** ♦ Purpose, range and types of users
Possible topics/activities	Creation of websites/multimedia products/blogs/wikis/databases etc to communicate information. For example: ♦ Photobook/Comicbook creation using captured images ♦ Creation of narrated/recorded tour GIS software/Google Earth ♦ Learning blog with text and images	Design and creation of websites/multimedia products/blogs/wikis/databases etc to store and communicate information. For example: Group of candidates blogging/commenting using existing blog(s) including: ♦ position/size of images ♦ editing and content management Selection and use of appropriate image software to complete image editing prior to upload to blog. Group collaboration on wiki to produce single wiki site created and edited by group on a topic of choice. Include images, text and other media elements.	Design and creation of websites/multimedia products/blogs/wikis/databases etc to store and communicate information. For example: Develop a blog site using categories, pages, tags, links and other features to provide structure to content. Develop blog using a variety of media types including: ♦ video ♦ audio ♦ image Investigation of blogging software features in simple terms: ♦ access to a range of features ♦ editing tools ♦ tools to allow sharing of content Design simple database using a single table and a variety of data types including: text, number, image to collect data for specific topic. Create a simple, interface to allow data to be viewed/searched/maintained using DBMS tools.	Design and creation of websites/multimedia products/blogs/wikis/databases etc to meet the needs of a user group. For example: Using a blog to create a structure and links using categories, pages, tags, links and other features. Adjust the blog user interface/theme with own menus, widgets and other user interface elements. Write or edit content using HTML view: ♦ common tags and attributes Develop blog using a variety of media types including: ♦ video ♦ audio ♦ image Investigation of blogging software design and implementation (including features for the following content related user types): ♦ administrator ♦ editor ♦ author ♦ subscriber Design database using at least two linked tables and a variety of data types including: text, number, image and object to collect data for specific topic. Create a simple interface to allow data to be viewed/searched/maintained/sorted using DBMS tools. Include a method of reporting the data.

Aspects of computational thinking: abstraction	Different forms of information: text, sound, graphics etc can be represented and stored in computers.	Different media types to represent different forms of information. Operations that can be performed on these. Different tools/software can be used depending on operations required.	Information can be structured and linked in different ways depending on purpose. Different tools are available depending on structure and links required.	Information can be structured and linked in different ways depending on purpose. Different tools are available depending on structure and links required.
Aspects of computational thinking: algorithmic thinking	Sequences of operations to create solution.	Sequences of operations to create solution.	Sequences of operations to create solution.	Creation of simple algorithms in the design and creation of simple scripts.
Aspects of computational thinking: decomposition			Creating a design based on available structure.	Creating a design based on available structure. Breaking problems into sub-problems (simple programs).
Aspects of computational thinking: generalisation			Ability to use same structure and links with different information. Scalability.	Ability to use same structure and links with different information. Use of scripting to enhance this. Scalability
Deterministic nature of computer	Output can be predicted.	Output can be predicted.	Predicting output. Testing. Rectifying errors.	Predicting output. Testing. Rectifying errors.

<table>
<tr><td colspan="5">Computer systems and architecture</td></tr>
<tr><td>Related experiences, outcomes and Assessment Standards</td><td>Third level experiences and outcomes</td><td>Fourth level experiences and outcomes</td><td>National 4 Outcomes and Assessment Standards</td><td>National 5 and Outcomes and Assessment Standards</td></tr>
<tr><td></td><td>Having gained knowledge and understanding of the components of a computer, I can make an informed choice when deciding on the system required for a specific purpose. (TCH 3-08c)</td><td>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)</td><td>(ISDD O2.2)
Consider a number of basic factors involved in the design and implementation of an information system by describing, in simple terms:
♦ Its hardware, software and connectivity requirements

(SDD O1.3)
Explain how simple programs work, drawing on understanding of basic concepts in software development by:
♦ Explaining how data and instructions are stored</td><td>(ISDD O2.2)
Consider the factors involved in the design and implementation of an information system by describing its:
♦ Implementation (hardware and software requirements, storage and connectivity)

(SDD O1.4)
Explain how simple programs work, drawing on understanding of basic concepts in software development by:
♦ Describing in simple terms how programs relate to low-level operations and structures

(SDD O3.4)
♦ Describing how high-level code is translated and executed</td></tr>
<tr><td>Possible topics/activities</td><td>Features of computer systems
Selecting hardware based on device function or observable performance:

Input, output, storage</td><td>Features of computer systems
Research emerging technology and describe impact or possible impact in the world of work, eg consider how mobile communications have impacted on the way we now work.

Input, output, storage, connectivity.

Increasing power of computer systems.

Increasing availability and volume of data.</td><td colspan="2">Features of computer systems
Input, output, processor, storage, connectivity, eg consider hardware specification of mobile device in these terms.

Operating system and application software.

Connected applications that use remote data sources, eg mobile apps which use cloud storage.</td></tr>
<tr><td></td><td></td><td></td><td>Need for translation
Concept that programs contain instructions encoded in binary.</td><td>Need for translation
Explanation of the need for translation.

Process of translation and translation software:
♦ compiler
♦ interpreter</td></tr>
<tr><td></td><td></td><td></td><td>Binary representation
Binary representation of data:
♦ character codes
♦ positive integers

Programs and data all stored as binary code.</td><td>Binary representation
Representation and use of variable in program:
♦ variable storage
♦ use of buses(read/write from/to memory)
♦ storage in processor (registers)

eg demonstration of low-level machine emulator (x84, 6502)</td></tr>
<tr><td>Aspects of computational thinking: abstraction</td><td>Simple computer model.</td><td>Simple computer model.</td><td>Use of binary to represent data and instructions. Interpretation of binary data depends on purpose (data or instruction).</td><td>High-level language, machine code.</td></tr>
<tr><td>Deterministic nature of computer</td><td colspan="4">A computer is a machine that carries out instructions to process information. Its behaviour depends on the information that is input and the instructions that are carried out.</td></tr>
</table>

Security considerations				
Related experiences, outcomes and Assessment Standards	**Third level experiences and outcomes**	**Fourth level experiences and outcomes**	**National 4 Outcomes and Assessment Standards**	**National 5 and Outcomes and Assessment Standards**
	By considering ways to protect technological devices, I can act safely and responsibly when selecting and using different technologies to communicate and collaborate. (TCH 3-08a)	I can compare different forms of security software to gain knowledge and understanding of their functions in protecting contemporary technologies. (TCH 4-08b)	**(ISDD O2.3)** **Consider a number of basic factors involved in the design and implementation of an information system by describing, in simple terms:** ♦ the security risks involved in digital communication	**(ISDD O2.3)** **Consider the factors involved in the design and implementation of an information system by describing its:** ♦ security risks and precautions
Possible topics/activities	Importance of protecting devices from third-parties. Methods include: ♦ PIN code ♦ username/password ♦ pattern matching ♦ face recognition Internet safety Using a variety of tools to communicate and collaborate: ♦ video/voice/instant messaging ♦ shared online documents using real-time collaboration	Investigation of the role of security software: ♦ virus protection ♦ firewall Study of possible threats to a computer user: ♦ viruses, worms, Trojans ♦ hacking	Study of possible threats to a computer user: ♦ viruses, worms, Trojans ♦ hacking Case study of common interception attacks such as man-in-the-middle and man-in-the-browser.	Case study of phishing attacks and their use to access personal information: ♦ Means of access: spoofing email or instant messaging ♦ Current targets: social networks, financial institutions ♦ Information theft: website forgery, Trojan delivery Case study of how cross-site scripting can be used to compromise websites and services. Requirement to 'sanitize' user input to prevent attacks. Encryption to reduce impact if data access is compromised.
Aspects of computational thinking: abstraction			How software and protocols can be subverted for other purposes.	Connectivity and spoofing. Encryption.
Aspects of computational thinking: algorithmic thinking			Steps used to carry out an attack.	Virus recognition. Understanding of weakness in programs and how these can be rectified.
Aspects of computational thinking: decomposition		Analysis of attack to identify nature of threat.	Analysis of attack to identify nature of threat.	Analysis of attack to identify nature of threat. Ability to analyse for potential security issues.
Aspects of computational thinking: generalisation	Using a variety of security tools to achieve the same level of protection.			Common methods of validating and sanitizing user input.
Deterministic nature of computer		Once effective protection in place, future specific threats are eliminated.	Once effective protection in place, future specific threats are eliminated.	Once effective protection in place, future specific threats are eliminated.

Impact of technological developments on society and environment				
Related experiences, outcomes and Assessment Standards	**Third level experiences and outcomes**	**Fourth level experiences and outcomes**	**National 4 Outcomes and Assessment Standards**	**National 5 and Outcomes and Assessment Standards**
	From my studies of technologies in the world around me, I can begin to understand the relationship between key scientific principles and technological developments. (TCH 3-01a)	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)	**(SDD O3.3)** **Produce a short factual report on a contemporary software-based application by:** ♦ Describing its impact on the environment or society	**(ISDD O2.4/5)** **Consider the factors involved in the design and implementation of an information system by describing its:** ♦ Legal implications ♦ Impact on the environment
Possible topics/activities		Investigation and discussion: Augmented reality: use of devices, wearable technology etc to enhance reality. GPS mobile devices and personal privacy: impact of location awareness for direct selling/advertising and personal safety. Simple overview of EPEAT environmental criteria and certification for computing device manufacturers.	Investigations and discussion: Possible software-based applications: social networking or instant messaging clients Investigation of EPEAT environmental criteria and certification for computing device manufacturers: ♦ levels of compliance ♦ impact on recycling and energy efficiency	Investigation of EPEAT criteria to develop understanding of: ♦ Reduction/elimination of environmentally sensitive materials ♦ Material selection ♦ Design for end of life ♦ Product longevity/life extension ♦ Energy conservation ♦ End-of-life management ♦ Corporate performance ♦ Packaging
Aspects of computational thinking: decomposition			Linking application functions/operations to programming constructs and data types.	
Aspects of computational thinking: generalisation		Common set of environmental standards can be applied across all manufacturers.	Common set of environmental standards can be applied across all manufacturers.	Common set of environmental standards can be applied across all manufacturers.