COMPUTING SUBJECT KNOWLEDGE AUDIT

Use this needs analysis to help self-assess and track your computing subject knowledge.

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
1 Computational thinking Define, explain and use these concepts with examples:			
Algorithm			
Program			
Decomposition			
Abstraction			
Generalization			
Logical reasoning			
Pseudocode			
Flowcharts			
2 Algorithms, programming and data structures Define and explain these concepts with examples:		Your links should cover examples of each concept when constructing algorithms and programming in at least 2 languages	
Sequencing			
Selection			
Repetition			
Procedures and functions			
Parameters			

Topic Area	Current Level of Understanding 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	Links to notes and resources	Teaching ideas and contexts
Lists and arrays			
Data types			
Variables			
Constants			
Operators: mathematical, string, relational and logical			
Variable scope			
Nested loops			
File handling			
Libraries			
2.1 Common algorithms Explain how these algorithms work in English and pseudocode:			
Insertion Sort			
Selection Sort			
Bubble Sort			
Quick Sort			
Merge Sort			
Linear Search			

(Continued)

(Continued)

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
Binary Search			
Explain and how a programmer might choose an algorithm for a given task or dataset. Explain how the performance or efficiency of algorithms can be described and evaluated e.g. using Big O notation.			
2.2 Debugging techniques Explain and use a range of debugging techniques for algorithms and programs including:			
Commenting			
Stepping			
Breakpoints			
Watching variables			
Trace dumps and tables			
Exception handling			
Interpreting error messages			
Understanding the difference between syntax, run-time and logical errors			
Selective code execution			

Topic Area	Current Level of Understanding 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	Links to notes and resources	Teaching ideas and contexts
Functional testing (especially boundary conditions)			
2.3 Error detection and testing Explain types of errors in data caused by user input or communication failure. Explain error detection techniques such as:			
Validation			
Verification			
Parity bits			
Checksums			
Repetition codes			
Explain strategies for testing programs and algorithms including:			
Logical testing (white box)			
Functional testing (black box)			
Integration testing			
End-user testing			
Derive functional test data for a program or algorithm using equivalence classes.			

(Continued)

(Continued)

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
Construct a trace table for an algorithm or program.			
2.4 Systems development lifecycles Describe and explain a range of systems development approaches, for example: Cyclical, waterfall, spiral, prototyping			
3 Computer architecture and data storage Define, explain and use these concepts:			
Computer			
CPU			
ALU			
Control unit			
Memory			
RAM			
ROM			
Bit			
Byte			

Topic Area	Current Level of Understanding 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	Links to notes and resources	Teaching ideas and contexts
Secondary storage			
Von Neumann architecture			
Operating system			
Moore's Law			
Data			
Information			
3.1 Operating systems Explain the tasks carried out by a typical operating system. Compare the characteristics of operating systems e.g. Windows and Android in terms of high level functionality.			
3.2 Physical architecture Explain the role of transistors in computers and how they can be connected to create logic gates. Explain how logic gates can be combined to perform useful tasks. Explain Boolean logic and derive logic tables for common logic gates. Explain how the use of transistors and logic gates gives rise to binary computation.			

(Continued)

(Continued)

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
3.3 Abstraction: 'levels' of programming languages Explain the difference between source code and object code (also known as machine code or executable code). Explain the concept of 'high level' and 'low level' programming languages and the differences between levels. Explain the role of compilers and interpreters and how they differ. Explain why executable code cannot usually be converted back into source code and why assembly language is an exception to this. Describe the fetch–execute cycle. Explain how machine code and assembly languages work and the relationship between them using a simplified example such as Little Man Computer. Write and interpret simple programs using a low level language such as assembly code/Little Man Computer.			
3.4 Number systems/data representation Explain how the binary number system can represent:			
Unsigned integers			
Signed integers			

Topic Area	Current Level of Understanding 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	Links to notes and resources	Teaching ideas and contexts
Long integers			
Fractions			
Characters			
Bitmaps			
Audio data			
Explain analogue-digital conversion, for example how sound sampling works and the parameters which determine the fidelity of such encoding e.g. frequency and bitrate.			
3.4 Number systems/data representation (cont.) Convert between binary, decimal and hexadecimal values.			
Explain the uses of hexadecimal as an intermediate number system.			
Perform simple binary mathematics and explain two's complement			
Explain the limitations of using binary representations e.g. overflow errors, rounding errors and fractional numbers.			

(Continued)

(Continued)

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
Explain how the same information can be encoded in different ways e.g. bitmap and vector graphics or sound waveforms and MIDI data.			
3.5 Data compression Explain common methods for data compression in simple terms, for example: Audio compression Video compression Zip compression Image compression Explain the difference between 'lossless' and 'lossy' compression.			
4 Networks and the Internet Define and explain:			
World Wide Web			
Internet			
URL			
Browser			
Data packets			

Topic Area	Current Level of Understanding 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	Links to notes and resources	Teaching ideas and contexts
Protocols			
Client–server models			
MAC address			
IP address			
Domain names			
Cookies			
Routing			
Web API			
Hyperlink			
HTML			
4 Networks and the Internet (cont.) Describe simple network topologies and identify their advantages and disadvantages. Explain in broad terms how data are transported in networks including the Internet.			
Explain a search engine strategy/algorithm to construct its page index and ranking. Explain strategies to construct search terms which improve pages returned in common search engines.			

(Continued)

(Continued)

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
5 Applications of computing Have competent skills and knowledge of a range of hardware and software including: Office software Collaboration and social networking software Web design software Image manipulation software Video production software Audio production software Physical monitoring and control applications			
5.1 Models and simulations In the context of computer models and simulations, define the terms:			
Model			
Simulation			
Rule			
Variable			
Assumption			
Explain with examples the relative advantages and disadvantages of models and simulations.			

Topic Area	Current Level of Understanding 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	Links to notes and resources	Teaching ideas and contexts
5.2 Relational database systems Explain the concepts of database files, records and fields.			
Explain how information can be structured in a relational database by using top down (entity relationship diagram) and bottom up (normalization) approaches. Explain the use of primary and foreign keys.			
Explain the terms data consistency, data redundancy, data integrity and data independence.			
Interpret and use simple SQL operations.			
6 Impact of computing – safety, legal and socio-economic issues Define and explain:			
Phishing			
Viruses			
Trojan horse			
Malware			
Identity theft			

(Continued)

(Continued)

Topic Area	**Current Level of Understanding** 3 = I could explain this to others 2 = I understand this concept 1 = I have little or no understanding of this concept	**Links to notes and resources**	**Teaching ideas and contexts**
Firewall			
Filtering software (blacklist and whitelist)			
Encryption			
Cyber bullying			
Digital footprint			
6.1 E-safety and security Identify threats to personal and professional safety online and explain e-safety steps to manage any risk.			
Explain steps to maintain data security, integrity and privacy.			
6.2 Legislation Explain legislation directly related to computing including the Data Protection Act, Computer Misuse Act and relevant copyright legislation (including the use of creative commons licences).			
6.3 Impact of computing Identify and consider social and ethical issues raised by the role of computers for individuals, organizations and society. Identify real-life computing contexts which exemplify the possibilities of technology, and provide children with a wide range of career role models related to the subject.			