**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 |
| --- | --- |
| **1. Computational Thinking**  **Define, explain and use these concepts with examples:**  **Algorithm** |  |
| **Program:** |  |
| **Decomposition:** |  |
| **Abstraction:** |  |
| **Generalisation:** |  |
| **Logical reasoning:** |  |
| **Pseudocode:** |  |
| **Flowcharts:** |  |
| **2. Algorithms, programming and data structures**  **Define and explain these concepts with examples.** |  |
| **Selection:** |  |
| **Repetition:** |  |
| **Procedures and Functions:** |  |
| **Parameters:** |  |
| **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 the how these algorithms work in English and pseudocode:**  **Insertion Sort** |  |
| **Selection Sort** |  |
| **Bubble Sort** |  |
| **Quick Sort** |  |
| **Merge Sort** |  |
| **Linear Search** |  |
| **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** |  |
| **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.** |  |
| **Construct a trace table for an algorithm or program** |  |
| **2.4 System Development Lifecycles**  **Describe and explain a range of systems development approaches for example:**  **Cyclical, waterfall, spiral, prototyping.** |  |
| **3 Computer Architecture & Data Storage**  **Define, explain and use these concepts:**  **Computer** |  |
| **CPU** |  |
| **ALU** |  |
| **Control Unit** |  |
| **Memory** |  |
| **RAM** |  |
| **ROM** |  |
| **Bit** |  |
| **Byte** |  |
| **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.** |  |
| **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** |  |
| **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 - eg overflow errors, rounding errors, and fractional numbers.** |  |
| **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 e.g.**  **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** |  |
| **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.** |  |
| **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.** |  |
| **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 (Normalisation) 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** |  |
| **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 licenses).** |  |
| **6.3 Impact of Computing**  **Identify and consider social and ethical issues raised by the role of computers for individuals, organisations 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.** |  |