

# Computer Science Subject Knowledge Audit

Throughout this course, you must demonstrate continuous improvement in your subject knowledge to meet the required standards. Strong expertise in your subject is essential for effective teaching. You will show your knowledge partly through lesson observations, but also by completing the subject knowledge audit below at each assessment point during your training year. The audit's red-amber-green rating of course content will map your progress. This live document will serve as a professional tool that you can build on throughout the early stages of your teaching career.

<b>RAG Rating Key</b>	
<b>S</b>	Secure knowledge = I have good pedagogical content knowledge and would be able to teach it
<b>D</b>	Developing knowledge = I understand it, I need to study it further to be able to teach it
<b>L</b>	Limited Knowledge = I have little knowledge and have not seen it taught

<b>Subject Knowledge</b>	<b>Pre-Course Knowledge</b>	<b>AP1 Autumn Term</b>	<b>AP2 Spring Term</b>	<b>AP3 Summer Term</b>	<b>Final Assessment Summer Term</b>	<b>Action Plan for Development</b>
<b>Dates</b>		18 November 24 to 2 December 24	3 March 25 to 24 March 25	28 April 25 to 12 May 25		
<b>National Curriculum</b>						
KS1 and 2 Computing National Curriculum						
KS3 and 4 Computing National Curriculum						
<b>Exam Specification</b>						
AQA GCSE Computer Science Specification						
OCR GCSE Computer Science Specification						
EdExcel GCSE Computer Science Specification						
<b>Algorithms</b>						
Abstraction						
Decomposition						

	Pre-Course Knowledge	AP1 Autumn Term	AP2 Spring Term	AP3 Summer Term	Final Assessment Summer Term	Action Plan for Development
Designing, creating and refining algorithms						
Searching and sorting algorithms						
Efficiency of algorithms						
<b>Programming</b>						
Data types						
Programming concepts (assignment, sequence, selection, iteration, subroutines)						
Input and output						
Arithmetic operators						
Relational operators						
Boolean (logical) operators						
Structured programming						
Classification of programming languages and translators						
Python syntax						
C# syntax						
<b>Data</b>						
Binary number system						
Hex number system						
Converting between number bases						
Binary arithmetic						
Units of data						
Representation of data (images, sound, text)						
Compression						
Encryption						
<b>Computer Systems</b>						
Hardware and software						
Software classification (system and application)						
Role of the operating system						
Boolean logic (logic circuits, Boolean expressions, truth tables)						

	Pre-Course Knowledge	AP1 Autumn Term	AP2 Spring Term	AP3 Summer Term	Final Assessment Summer Term	Action Plan for Development
Systems architecture - stored program concept						
Fetch-Execute cycle						
CPU performance						
Memory (RAM & ROM)						
Storage (magnetic, optical, solid state)						
Embedded systems						
<b>Networking</b>						
Types of networks (PAN, LAN, WAN)						
Network topologies (Star, bus, mesh)						
Wired and wireless networks						
TCP/IP stack						
Application layer protocols						
Addressing (IP & MAC)						
<b>Impacts</b>						
Ethical issues						
Legal issues						
Cultural issues						
Environmental issues						
Privacy issues						
Relevant laws						
<b>Cyber Security and Online Safety</b>						
Threats to computer systems						
Identification and prevention of vulnerabilities						
Safe online behaviour						
<b>ICT Skills</b>						
Use computers to undertake creative projects						
Use, create and edit digital artefacts in projects						
Consider appropriateness and trustworthiness of digital artefacts when using and creating						
Consider usability of digital artefacts when using and creating						