AS and A LEVEL COMPUTER SC

Scheme of work teaching hours	
Teaching weeks:	35
Teaching hours/week:	5
Total teaching time:	175
Scheme teaching time:	**121**

Please read guidance notes on right when planning your scheme of work

Teaching Hours	Topic
6	Structure and Function of Processor
9	Types of Processor
8	Input, Output and storage

9	Systems Software (A Level) Operating Systems (AS Level)
8	Applications Generation
8	Software Development
20	Types of Programming Language
7	Compression, Encryption and Hashing
6	Databases

5	Networks
7	Web Technologies
8	Data Types
10	Boolean Algebra
5	Computing related legislation
5	Moreal and Ethical Issues

Guidance to teachers

Allocated teaching time is assuming that you are teaching the full A Level, and thus need to include the Component 03 Programming Project.

If solely teaching AS Level, you can adjust teaching time accordingly, as you will not need to factor in the time for the Programming Project, but will need to deliver the Component 02 content.

AS Level is viewed as a ***one year course*** *and fits well with A Level if co-teaching. Stretching AS over two years may make co-teaching significantly more difficult to coordinate.*

Centres may wish to deliver AS in Year 12 and A Level in Year 13. Candidates likely to move on to Year 13 should be studying programming throughout the first year to consolidate their skills. Solely delivering practical programming skills in Year 13 may hinder their progress within the Component 03 Programming Project unit.

Sub Topic
The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control
The Fetch-Decode-Execute Cycle; including its effects on registers.
The factors affecting the performance of the CPU: clock speed, number of cores, cache.
The use of pipelining in a processor to improve efficiency
Von Neumann, Harvard and contemporary processor architecture.
The differences between and uses of CISC and RISC processors.
GPUs and their uses (including those not related to graphics).
Multicore and Parallel systems.
How different input, output and storage devices can be applied to the solution of different problems.
The uses of magnetic, flash and optical storage devices.
RAM and ROM.
Virtual storage.
The need for, function and purpose of operating systems.
Memory Management (paging, segmentation and virtual memory).
Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle.

Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time.

Distributed, embedded, multi-tasking, multi-user and Real Time operating systems.

BIOS.

Device drivers.

Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within another.

The nature of applications, justifying suitable applications for a specific purpose.

Utilities.

Open source vs. closed source.

Translators: Interpreters, compilers and assemblers.

Stages of compilation (lexical analysis, syntax analysis, code generation and optimisation).

Linkers and loaders and use of libraries.

Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development.

The relative merits and drawbacks of different methodologies and when they might be used.

Writing and following algorithms.

Different test strategies, including black and white box testing and alpha and beta testing

Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation.

Need for and characteristics of a variety of programming paradigms.

Procedural languages:

- program flow
- variables and constants
- procedures and functions
- arithmetic, Boolean and assignment operators
- string handling
- file handling.

Assembly language (including following and writing simple programs with the Little Man Computer instruction set).

Modes of addressing memory (immediate, direct, indirect and indexed).

Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism.

Lossy vs. Lossless compression.

Run length encoding and dictionary coding for lossless compression.

Symmetric and asymmetric encryption.

Different uses of hashing.

Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing.

Methods of capturing, selecting, managing and exchanging data.

Normalisation to 3NF.

SQL – Interpret and modify.

Referential integrity.

Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy.

Characteristics of networks and the importance of protocols and standards.

The internet structure:
• The TCP/IP Stack.
• DNS
• Protocol layering.
• LANs and WANs.
• Packet and circuit switching.
Network security and threats, use of firewalls, proxies and encryption.
Network hardware.
Client-server and peer to peer.

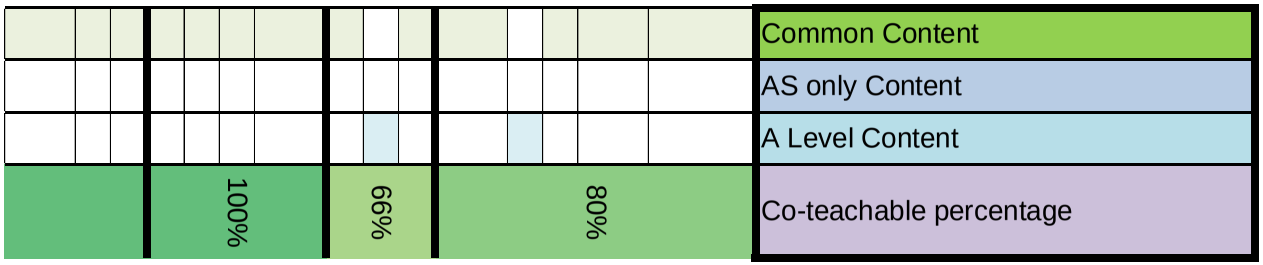
HTML, CSS and JavaScript.
Search engine indexing.
PageRank algorithm.
Server and client side processing.

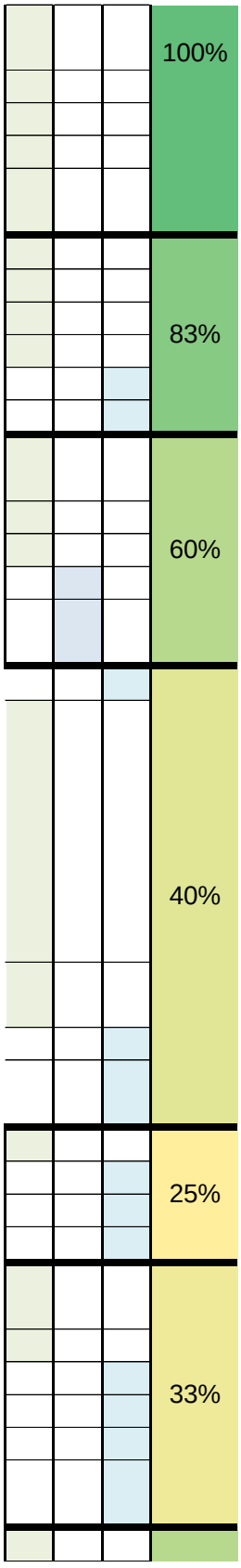
Primitive data types, integer, real/floating point, character, string and Boolean.
Represent positive integers in binary.
Use of sign and magnitude and two's complement to represent negative numbers in binary.
Addition and subtraction of binary integers.
Represent positive integers in hexadecimal.
Convert positive integers between binary hexadecimal and denary.
Representation and normalisation of floating point numbers in binary.
Floating point arithmetic, positive and negative numbers, addition and subtraction.
Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR.
Positive and negative real numbers using normalised floating point representation
How character sets (ASCII and UNICODE) are used to represent text.

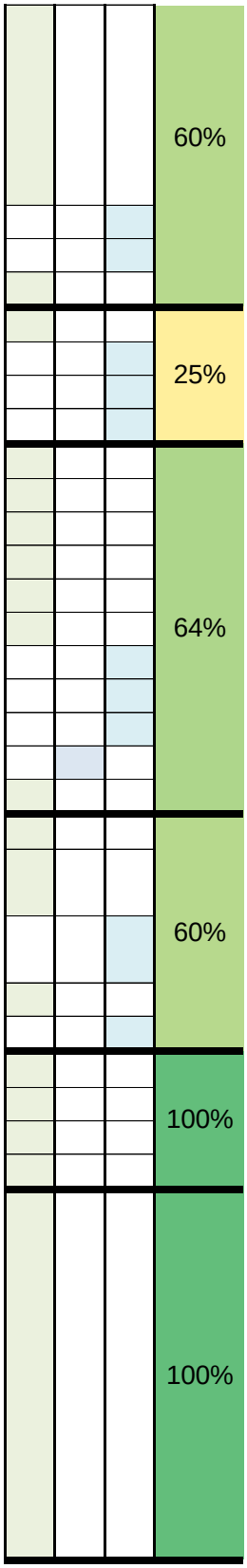
Define problems using boolean logic.
Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions
Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.
Using logic gate diagrams and truth tables.
The logic associated with D type flip flops, half and full adders.

The Data Protection Act 1998.
The Computer Misuse Act 1990.
The Copyright Design and Patents Act 1988.
The Regulation of Investigatory Powers Act 2000.

The individual moral, social, ethical and cultural opportunities and risks of digital technology:
• Computers in the workforce.
• Automated decision making.
• Artificial intelligence.
• Environmental effects.
• Censorship and the Internet.
• Monitor behaviour.
• Analyse personal information.
• Piracy and offensive communications.
• Layout, colour paradigms and character sets.


Common Content
AS only Content
A Level Content
100%
66%
80%
Co-teachable percentage


100%
83%
60%
40%
25%
33%


60%
25%
64%
60%
100%
100%

Resource Links
Online Delivery Guide: Structure and function of the processor Topic Exploration Pack - Teacher Instructions: Structure and function of the Processor Learner Activity: Structure and Function of the Processor
Online Delivery Guide: Types of Processor Topic Exploration Pack - Teacher Instructions: Types of Processor Topic Exploration Pack - Learner Activity: Types of Processor
Online Delivery Guide: Input, output, storage
Online Delivery Guide: Systems Software Topic Exploration Pack - Teacher Instructions: Systems Software Topic Exploration Pack - Learner Activity: Systems Software

Topic Exploration Pack - Teacher Instructions: Application Generation Topic Exploration Pack - Learner Activity: Application Generation
Online Delivery Guide: Software Development Topic Exploration Pack - Teacher Instructions: Software Development Topic Exploration Pack - Learner Activity: Software Development
Online Delivery Guide: Types of Programming Language Topic Exploration Pack - Learner Activity: Types of Programming Language
Online Delivery Guide: Compression, Encryption and Hashing Topic Exploration Pack - Teacher Instructions: Compression, Encryption and Hashing *For AS LEVEL Lossy v Lossless Compression is part of 1.3.3 Web Technologies*
Online Delivery Guide: Software Development Topic Exploration Pack - Teacher Instructions: Software Development Topic Exploration Pack - Learner Activity: Software Development
Online Delivery Guide: Networks

Topic Exploration Pack - Teacher Instructions: Networks
Online Delivery Guide: Web Technologies Topic Exploration Pack - Teacher Instructions: Web Technologies Topic Exploration Pack - Learner Activity: Web Technologies
Online Delivery Guide: Data Types Topic Exploration Pack - Teacher Instructions: Data Types Topic Exploration Pack - Learner Activity 1: Data Types Topic Exploration Pack - Learner Activity 2: Data Types Topic Exploration Pack - Learner Activity 3: Data Types
Online Delivery Guide: Boolean Algebra Topic Exploration Pack - Teacher Instructions: Boolean Algebra Topic Exploration Pack - Learner Activity 1: Boolean Algebra Topic Exploration Pack - Learner Activity 2: Boolean Algebra Topic Exploration Pack - Learner Activity 3: Boolean Algebra
No current supporting resources for this unit
No current supporting resources for this unit