A Level Computer Science Curriculum Overview (Year 12)

Introduction
The course very naturally falls into two parts being tested through a more practical Paper 1 and more theory-based paper 2. Although there are activities to supplement and support both parts of the course (especially where there is overlap anyway) the split means normally one teacher will focus on developing the students' programming skills whilst the other focuses on developing written techniques. Even if taught by 1 teacher time will be split to ensure both areas develop concurrently. teacher 1 (focus on sections 1-4 of the specification, more practical, paper 1) teacher2 (focus on sections 5-9 of the specification, more theory, paper 2) AQA specification: AQA | Computer Science | A-Level | A-level Computer Science Extension links Computerphile - YouTube W3Schools Online Web Tutorials

Term 1 Half Term 1 – teacher 1				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Fundamentals of programming	**Knowledge**: Data types (integer, real/float, Boolean, character, string, date/time, records, arrays). Programming concepts (variable declaration, constant declaration, assignment, iteration, selection, subroutine (inc. parameters, returning values and global/local variables)). Arithmetic operations (inc integer division, exponentiation, rounding, truncation). Relational operations (equal to, not equal to, less than, greater than, less than or equal to, greater than or equal to). Boolean operations (NOT, AND, OR, XOR). String-handling operations. Random number generation. Structured programming **Understanding:** Students will understand how the above items can be used together to create programs that solve problems. They will understand how to develop subroutines that pass parameters and return values . Understand the structured approach to program design and construction and be able to construct and use hierarchy charts when designing programs. **Skills:** Developing and formatting readable code	Past paper questions used in lessons as formative assessment: Section 1 practice Formal test used as summative assessment: Section 1 Assessment – Noughts and Crosses Skeleton Program	A series of Python files can be used to teach and practice this section of the course. Practice questions used to consolidate concepts. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p2-34 Resources – outline Powerpoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Practice skeleton program (as part of assessment) – Noughts and Crosses Many Python files available to demonstrate and practise the concepts here.

Term 1 Half Term 1 – teacher 2				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Data representations	**Knowledge**: Number systems N, Z, Q, R. Bases 2, 10, 16. Units to tebi. Two's complement binary. ASCII and Unicode. Parity bits, majority voting, check digits. Bitmaps. Analogue/digital conversion of sound. MIDI. Run-length encoding and dictionary-based methods. Caesar and Vernam ciphers. **Understanding:** Why hexadecimal is used as a shorthand for binary. How the same byte could represent many different things. Why unicode was introduced. How an ADC works. The difference between lossless and lossy compression and the advantages of each. **Skills:** Covert between different number bases. Calculation involving binary (without converting to decimal)	Past paper questions used in lessons as formative assessment: Section 5 part 1 practice Section 5 part 2 practice Formal homeworks used as summative assessment: Section 5 part 1 assessment Section 5 part 2 assessment	PowerPoints used to cover key learning points include activities for students to practice. Past exam questions then used to consolidate and check understanding AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p182-228 Resources – outline Powerpoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Practical tasks: Converting bases Error checking Skeleton – Secret Messages

Term 1 Half Term 2 – teacher 1				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Data structures	**Knowledge**: Be familiar with the concept of data structures and arrays (single and multi-dimensional). How to read/write from a text file and a binary (non-text) file. **Understanding:** How a 2D array can be used to represent a table, matrix or grid. **Skills:** Developing and formatting readable code, including the use of arrays. Continuing to build programming skills through practical experiences, including the use of a previous skeleton program.	Past paper questions used in lessons as formative assessment: Section 2 practice Formal test used as summative assessment: Section 2 Assessment – Plant Sim Skeleton Program	A series of Python files can be used to teach and practice this section of the course. Practice questions used to consolidate concepts. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p50-55 (AS parts only) Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Skelton program – Plant Sim (as part of assessment) Many Python files available to demonstrate and practise the concepts here.

Term 1 Half Term 2 – teacher 2				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Computer systems	**Knowledge**: Classification of hardware and software. Role of an OS. Classification of high- and low-level languages. Compilers and interpreters. Bytecode. Logic gates. Basics of Boolean algebra. **Understanding:** the need for, and attributes of, different types of software (OS, utility, library, translator). Advantages and disadvantages of high- and low-level languages. Explain the differences between compilers and interpreters and situations in which each is appropriate. **Skills:** Complete a truth table for a given logic circuit. Write a Boolean expression for a given logic gate (and vice versa). Use Boolean identities and De Morgan's laws to manipulate and simplify Boolean expressions.	Past paper questions used in lessons as formative assessment: Section 6 part 1 practice Section 6 part 2 practice Formal homeworks used as summative assessment: Section 6 part 1 assessment Section 6 part 2 assessment	PowerPoints used to cover key learning points include activities for students to practice. Past exam questions then used to consolidate and check understanding AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p230-264 Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Practical tasks: Bitwise operators

Term 2 Half Term 1 – teacher 1				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Systematic approach to problem solving	**Knowledge**: The five stages of software development: Analysis, Design, Implementation, Testing, Evaluation (ADITE) **Understanding:** The concepts of normal, erroneous and boundary data for testing. **Skills:** Continued development of programming skills, including the use of a new skeleton program (note that although this section can be tested explicitly in AS exams, it is examined through the NEA project for the full A level)	Past paper questions used in lessons as formative assessment: Section 3 practice Formal test used as summative assessment: Section 3 Assessment – Battleships Skeleton Program	Students may be given the opportunity to complete a mini projects to experience the ADITE process in practice. This can lead on to the NEA project required in Year 13. Practice questions used to consolidate concepts. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p408-416 Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Skelton program – Battleships (as part of assessment)

Term 2 Half Term 1 – teacher 2				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Computer organisation and architecture	**Knowledge**: Basic internal components of a computer system (processor, main memory, buses). Harvard and von Neumann architectures. The stored program concept. The Fetch-Execute cycle and the role of registers within it. The processor instruction set. Immediate and direct addressing. Factors affecting processor performance. Assembly code operations. External hardware devices (barcode reader, digital camera, laser printer, RFID). Secondary storage devices (hard disk, optical disk, soli-state drive) **Understanding:** How the different bus widths affect system performance. The use of buses within the Fetch-Execute cycle. Compare and contrast the different external hardware for use in a given context. **Skills:** Be able to understand and write simple programs using the standard AQA assembly language.	Past paper questions used in lessons as formative assessment: Section 7 part 1 practice Section 7 part 2 practice Formal homeworks used as summative assessment: Section 7 part 1 assessment Section 7 part 2 assessment Exam Conditions January mock covers first terms work for teacher 2.	PowerPoints used to cover key learning points include activities for students to practice. Past exam questions then used to consolidate and check understanding AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p266-299 Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments

Term 2 Half Term 2 – teacher 1				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Theory of computation	**Knowledge**: Definitions of: algorithm, abstraction (representational, by generalisation or categorisation, procedural, functional), information hiding, data abstractions, problem abstraction/reduction, decomposition, composition, automation **Understanding:** Develop ability to break a problem down into its component parts. **Skills:** Develop and check solutions to simple logic problems	Past paper questions used in lessons as formative assessment: Section 4 practice Formal test used as summative assessment: Section 4 Assessment – current Skeleton Program	Practice questions used to consolidate concepts. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p134-149 (AS sections only) Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Skelton program – Use the current years AS program if there are any AS students, else can use Morse Code (as part of assessment)

Term 2 Half Term 2 – teacher 2				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Consequences of computing	**Knowledge**: Awareness of current individual (moral), social (ethical), legal and cultural opportunities and risks of computing. Awareness of some relevant laws (although specifics not required) such as the Data Protection Act and the Copyright, Designs and Patents Act. **Understanding:** Be able to categorise issues into the (sometimes overlapping) areas of individual, social, legal and cultural. **Skills:** Being able to analyse a given situation and apply typical issues (job creation/destructions, privacy, copyright, fault, access) to the specific situation being discussed. A standard approach is to 'brainstorm' the issues and then practice turning that into a longer written response.	Past paper questions used in lessons as formative assessment: Section 8 practice Formal homeworks used as summative assessment: Section 8 assessment	A variety of topics relating to this topic will be discussed. The students will then research and produce their own presentations of related topics of interest to them. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p300-309 Recent news stories involving the development and use of technology (eg smart cities; use of AI/AR/VR in fields such as recruitment and medicine) Resources – outline PowerPoints with suggested examples and scaffolding activities. Practice questions (from past exams) and section assessments Black Mirror (tv series)

Term 3 Half Term 1 – teacher 1				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Skeleton program preparation for mock	**Knowledge**: consolidation and development of all knowledge from 'Teacher1' sections above **Understanding:** consolidation and development of all understanding from 'Teacher 1' sections above **Skills:** consolidation and development of all skills from 'Teacher 1' sections above	Year 12 Mock Paper 1	Practice questions used to consolidate concepts. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Practice questions (from past exams) for section A Programming Challenges for end of Section A Practice theory questions on the skeleton program for Section B – PowerPoint with suggestions and examples Practice practical changes to the skeleton program for Section C (can also see AQA wikibooks page)

Term 3 Half Term 1 – teacher 2				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Communication and Networking	**Knowledge**: Serial and parallel transmission. Synchronous and asynchronous transmission. Definitions of: baud rate, bit rate, bandwidth, latency, protocol. Physical star, logical bus topology. Peer-to-peer and client-server networking. Wireless networking concepts: WiFi, NIC, WAP, security issues, CSMA/CA, RTS/CTS, SSID **Understanding:** Difference between bit rate and baud rate, and the relationship to bandwidth. How a topologys physical layout can be different to it's logical functionality. **Skills:** Calculate bit rate from baud rate and bits per signal. Draw and identify bus and star networks.	Past paper questions used in lessons as formative assessment: Section 9 part 1 practice Section 9 part 2 practice Formal homeworks used as summative assessment: Section 9 part 1 assessment Section 9 part 2 assessment	PowerPoints used to cover key learning points include activities for students to practice. Past exam questions then used to consolidate and check understanding AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p310-325 Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments

Term 3 Half Term 2 – teacher 1				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Programming NEA - Introduction	**Knowledge**: Be familiar with the key concepts: class, object, instantiation, encapsulation, inheritance, aggregation, composition, polymorphism, overriding. Use of 'ADITE' for a project. **Understanding:** The object-oriented paradigm and why it is used. Be aware of the principles: encapsulate what varies, favour composition over inheritance, program to interfaces, not implementation. **Skills:** Experience of programming in object-oriented programming paradigm involving: abstract, virtual and static methods, inheritance, aggregation, polymorphism, public/private/protected specifiers. Be able to draw and interpret class diagrams.	Past paper questions used in lessons as formative assessment: A Section 1 practice Formal test used as summative assessment: A Section 1 Assessment Students should also submit the Analysis section of their NEA for checking that they are appropriate and of A level standard. Note only general feedback can be given however.	A series of Python files can be used to teach and practice this section of the course relating to OOP and Classes. Examples of old NEA projects used to highlight requirements of this element of the course. Practice questions used to consolidate concepts. AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p25-49 Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments Many Python files available for introducing the various concepts and giving examples (eg of composition vs inheritance)

Term 3 Half Term 2 – teacher 2				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**Evaluation and assessment methods**	**Implementation**	**What additional resources are available?**
Databases and SQL	**Knowledge**: Relational databases and entity-relationship (ER) diagrams. Definitions of: entity, attribute, primary key, composite primary key, foreign key, normalisation. Use of SQL to retrieve, update, insert and delete data. Use of SQL to define a database table. Know how concurrent access to a database can be controlled to preserve the integrity of the database. **Understanding:** Why databases are normalised. Why concurrent access can cause issues and how to they can be prevented/resolved. **Skills:** Produce a data model from given data requirements for a simple scenario involving multiple entities. Be able to identify why given relations are not normalised, and be able to normalise given relations to third normal form	Past paper questions used in lessons as formative assessment: A Section 10 practice Formal test used as summative assessment: A Section 10 Assessment	PowerPoints used to cover key learning points include activities for students to practice. Past exam questions then used to consolidate and check understanding AQA A level Computer Science Text Book is largely used for consolidation outside of lessons	Textbook p364-393 Resources – outline PowerPoints with suggested examples and scaffolding activities Practice questions (from past exams) and section assessments For practising SQL: https://www.w3schools.com/ Python files: - Using sqlite - Using sqlite advanced