# A Level Computing Scheme of Work

Board: OCR

Course codes: H046 (AS Level), H446 (A Level)

Note: Year 1 covers the AS Level material

<table>
<tr><th>Year 1</th><th>Year 2</th></tr>
<tr><td colspan="2">Autumn Term</td></tr>
<tr><td>First Half</td><td></td></tr>
<tr><td>Unit 1.4.1 – Data Types<br>Key Skills: Developing an understanding of number systems in general and Decimal, Binary, Hexadecimal and Octal specifically.<br>Cultural Capital: Feeds directly into the study of Pure and Applied Mathematics.<br><br>Unit 1.4.2 – Data Structures<br>Key Skills: Understanding the functions of Queues, Stacks, Lists and Binary Trees, their differences and uses.<br>Cultural Capital: Supports the learning of computer programming and the need for preparation and planning in the solution of a problem.<br><br>Ongoing: Python programming</td><td>Non-Examination Assessment (NEA) preparation and commencement<br>Unit 1.4.2 – Data Structures<br>Key Skills: Understand and use Directed and Undirected Graphs. Scan binary trees using Inorder, Preorder and Postorder scans. Be able to convert between Inorder and Postorder (RPN) expressions. Be aware that the term ‘tree’ is used more loosely to mean any kind of branching structure.<br>Cultural Capital: Awareness of Preorder and Postorder (RPN) expressions shows that mathematics can be written without brackets, and that CPUs use RPN.<br><br>Ongoing: NEA project<br>Key Skills: The NEA draws together all strands of the course into a single piece of work. The student has to apply an agile development methodology to the system life cycle in order to produce a programmatic solution to a problem of their choice.<br>Cultural Capital: There is an opportunity here for the student to undertake a piece of work that they can fully take ownership of. Students are encouraged to ask family, friends and teachers if they have a task within their business, interests or teaching that could be solved programmatically so that the student is developing something that is actually solving a genuine real-world problem and will be employed by the end user.</td></tr>
<tr><td>Second Half</td><td></td></tr>
<tr><td>Unit 1.4.2 – Data Structures<br>Key Skills: See above.<br>Cultural Capital: See above.<br><br>Unit 1.4.3 – Boolean Algebra<br>Key Skills: Developing an understanding of Boolean logic, its purpose in circuit design and methods of simplifying circuits.<br>Cultural Capital: Directly supports the study of Logic Systems and Electronics. Simplification of circuitry encourages consideration of the need to</td><td>Unit 1.4.2 – Data Structures<br>Key Skills: See above.<br>Cultural Capital: See above.<br><br>Unit 1.4.3 – Boolean Algebra<br>Key Skills: Understand Half and Full Adder circuits and how they are used to perform Binary mathematics. Understand Karnaugh Maps and how they are used to simplify electronic circuits.</td></tr>
</table>

minimise the consumption of resources and power.

Unit 1.1.1 – Structure and function of the processor

**Key Skills:** Understanding the components within a computer's Central Processing Unit and how they interact with each other during the Fetch-Decode-Execute cycle.

**Cultural Capital:** Encourages logical thinking; that electronic devices perform as they do for a reason and that things such as errors do not 'just happen'. Reinforces the need for granular thinking and the ability to express complex concepts in a logical, detailed and accurate fashion, which feeds into other scientific studies.

Unit 1.1.2 – Types of processor

**Key Skills:** Understanding the similarities and differences between CISC and RISC processors, and multicore and parallel processing systems.

**Cultural Capital:** Encourages learners to think about the history of computing and the concepts proposed by people such as Von Neumann, and to think about how the development of more efficient processors reduces energy consumption and the mining and refining of raw materials including rare earth metals, which feeds into Geography, Geology, Ecology, Philosophy and Ethics.

Unit 1.1.3 – Input, output and storage

**Key Skills:** Developing an understanding of input, output and storage devices used in computing systems, their purpose, use and capacities, including devices to aid the disabled to use computers effectively.

**Cultural Capital:** Learners consider aspects of Health and Safety that arise from day-to-day computer use and steps that can be taken to reduce / eradicate them, such as eye strain, headaches and Carpal Tunnel Syndrome. Learners develop an appreciation of the working environment when using computer systems and employers' Health and Safety obligations. Develops an appreciation that computer systems are for the use of all and the measures that can be taken to ensure that the disabled are not disenfranchised.

Ongoing: Python programming

**Cultural Capital:** Directly supports the study of Logic Systems and Electronics. Simplification of circuitry encourages consideration of the need to minimise the consumption of resources and power.

Unit 1.1.2 – Types of processor

**Key Skills:** Understand the purpose of a GPU in a computer, and that GPUs can be used for other SIMD applications.

**Cultural Capital:** Develop an awareness that there are specialist computing systems making heavy use of GPUs in Single Instruction Multiple Data scenarios, including but not limited to modelling physical systems, audio processing, breaking passwords, machine learning and bitcoin mining.

Unit 1.2.2 – Applications generation

**Key Skills:** Understand the similarities and differences between Interpreters, Assemblers and Compilers and where each is used. Understand the steps involved in compilation. Understand what Linkers and Loaders are and how they can be used to create reusable code.

**Cultural Capital:** By looking at the differences between Interpreters, Assemblers and Compilers, learners develop an awareness of where they are used and why, particularly in websites. Studying how Abstract Syntax Trees are created demonstrates a recursive algorithm in action.

Unit 1.2.3 – Software development

**Key Skills:** Understand the stages of the system life cycle and what is involved in each. Understand the different life cycle methodologies and the project natures to which each is suited.

**Cultural Capital:** Understanding that all solutions of any nature – not just computer solutions – have a life cycle can be applied to any problem solving scenario. Appreciating the necessity of going through all the stages of the life cycle methodology used and for thorough documentation is a transferable skill that can be used in a multitude of other areas.

Unit 1.2.4 – Types of programming language

**Key Skills:** Understand the differences between procedural languages, assembly languages and object-oriented languages. Understand the different modes of memory addressing and why they are used.

**Cultural Capital:** Knowing that there are many types of language available to programmers gives learners an appreciation of the fact that

| | |
|---|---|
| | there are other languages other than those that they have learned. That there are different languages for different problem scenarios develops an understanding that problem solving means providing the best solution for the circumstances, rather than trying to shoehorn all solutions into a favoured framework which may be inappropriate or inefficient.<br><br>Ongoing: NEA project<br>**Key Skills:** See above.<br>**Cultural Capital:** See above. |
| Spring Term | |
| First Half | |
| Unit 1.2.1 – Operating systems<br>**Key Skills:** Awareness that the PC Operating Systems used at home, in school or at work are a small subset of the types used on mainframes in larger environment. Select the appropriate operating system according to processing requirements.<br>**Cultural Capital:** Learners develop an awareness that large organisations use mainframe computers for their processing needs and that these are very expensive pieces of equipment that need to work continually, the exact nature of how they work determined by the time of day and processing requirements.<br><br>Unit 1.2.2 – Applications generation<br>**Key Skills:** Learners appreciate that user-facing software (applications) and machine-facing software (utilities) are both programs, and that there are many more applications than Office and Graphics programs.<br>**Cultural Capital:** Learners develop an awareness that large organisations use mainframe computers for their processing needs and that these are very expensive pieces of equipment that need to work continually, the exact nature of how they work determined by the time of day and processing requirements.<br><br>Unit 1.2.3 – Introduction to programming<br>***This is identical to AS Unit 2.2.2***<br>**Key Skills:** Learners formalise what they have found out about sequence, iteration and branching naturally as part of the Python studies. Understand the differences between the various iteration and branching structures.<br>**Cultural Capital:** Learners appreciate that solutions to most complex problems can be represented as pseudocode, whether the problem will be solved using IT or not. Learners develop the skills to express the steps involved in developing solutions in great detail, a | Unit 1.2.4 – Types of programming language<br>**Key Skills:** See above.<br>**Cultural Capital:** See above.<br><br>Unit 1.3.1 – Compression, encoding and hashing<br>**Key Skills:** Understand the importance of compression, encoding and hashing in data transmission in terms of reducing transmission times and increasing security. Know the difference between lossy and lossless compression and where the use of each is acceptable. Understand the consequences of lossy compression.<br>**Cultural Capital:** An awareness of how and why encryption and hashing are used in data transmission develops of an awareness of how insecure some data transmission methods that are used every day can be and how easily data can be intercepted. This in turn develops a greater degree of caution as to what information might wish to share over differing platforms given their relative security. Understanding the effects of lossy compression gives a greater awareness of what the likely effects will be when, for instance, changing the bitrate of a sound file, the frame rate of a video or the resolution of an image.<br><br>Unit 1.3.2 – Networks<br>**Key Skills:** Understand the different network security threats and how they work, and the measures that can be taken to prevent them entering a computer system. Know the devices used to create modern networks, the purpose of each and how they interact.<br>**Cultural Capital:** This Unit encourages learners to take a hard look at their own home networks and smart devices, and to reflect on whether or not they are as secure as they could be, while also offering possible remedial actions. Understanding network hardware enables learners to make informed decisions when |

| | |
|---|---|
| transferable skill that can be applied to many other areas.<br><br>Unit 1.3.1 – Databases<br>**Key Skills:** Understand the terminology used in database design and be able to apply it to a conceptual data model. Be able to analyse a data processing problem, Atomise the data, put it in First, Second and Third Normal and build a relational database to store it using Access.<br>**Cultural Capital:** Learners appreciate the problems caused by inefficient and poorly designed data management systems, and the cost and data security issues that result from such systems. This feeds directly in to Business Studies, Sociology, Philosophy and Ethics. Learners develop the skills to forward plan in great detail, a transferable skill that can be applied to many other areas.<br><br>Ongoing: Python programming | installing or expanding their own home networks.<br><br>Unit 1.3.4 – Web technologies<br>***A Level Only***<br>**Key Skills:** Understand that a web page may use several different languages simultaneously to achieve its purpose. Understand the basics of HTML, CSS Stylesheets, JavaScript and php, and their purpose in the functionality of well-designed web pages.<br>**Cultural Capital:** By understanding how the different web technologies interact to make web pages work, an awareness is developed of what information is held client side and what information is stored server side. This informs as to where vulnerabilities in web design lie, and also reinforces earlier studies about databases. The use of CSS Stylesheets further reinforces the desirability in general of making computer code reusable when developing solutions.<br><br>Ongoing: NEA project<br>**Key Skills:** See above.<br>**Cultural Capital:** See above. |
| Second Half | |
| Unit 1.3.1 – Databases<br>**Key Skills:** See above.<br>**Cultural Capital:** See above.<br><br>Ongoing: Python programming | Unit 2.1.1 – Programming Techniques<br>**Key Skills:** Understand the meaning of Iteration and Recursion. Be able to define both iterative and recursive solutions to the same problem where appropriate.<br>**Cultural Capital:** Learners understand why recursive routines they have encountered before are written that way. A stronger appreciation of the use of the stack data structure to 'remember' the state of a program at stages during its running is developed.<br><br>Unit 2.1.5 – Thinking Concurrently<br>**Key Skills:** Understand the concepts of parallel processing and threading, and that these methods are not suited to all problems.<br>**Cultural Capital:** Thinking concurrently is a decision tool that can be used in any problem solving or system development situation, regardless of whether the solution is computer based or not.<br><br>Unit 2.2 – Computational Methods<br>**Key Skills:** Understand how divide and conquer, modular design and abstraction are used in solution development, and explore scenarios in which they are used. Understand where backtracking, anomaly detection, heuristics, performance modelling and visualisation are |

| | |
|---|---|
| | incorporated into solutions and explore scenarios where they might be used.<br>**Cultural Capital:** The concepts here are decision tools that can be used in any problem solving or system development situation, regardless of whether the solution is computer based or not.<br><br>Unit 2.3 – Algorithms<br>**Key Skills:** Understand how Big O notation can be used to determine algorithm complexity and scalability. Understand how Dijkstra's Algorithm and the A* Algorithm can be used to calculate the shortest path between two nodes of a weighted graph and explore uses, e.g. in route planning.<br>**Cultural Capital:** The concepts here are design tools that can be used in other problem solving or system development situations, regardless of whether the solution is computer based or not.<br><br>Ongoing: NEA project<br>**Key Skills:** See above.<br>**Cultural Capital:** See above. |
| Summer Term | |
| First Half | |
| Unit 1.3.1 – Databases<br>**Key Skills:** See above.<br>**Cultural Capital:** See above.<br><br>Unit 1.3.2 – Networks<br>**Key Skills:** Understand different network topologies, client-server and peer-to-peer networks. Understand the OSI 7 layer model and its relationship to TCP/IP.<br>**Cultural Capital:** Learners appreciate that the type of network used depends on the nature of the networking required. Understanding that the OSI exists instils an awareness of why standards are needed in communications and other technologies.<br><br>Unit 1.3.4 – Web technologies<br>***AS Level Only***<br>**Key Skills:** Understand that a web page may use several different languages simultaneously to achieve its purpose. Understand the basics of HTML, CSS Stylesheets, JavaScript and php, and their purpose in the functionality of well-designed web pages.<br>**Cultural Capital:** By understanding how the different web technologies interact to make web pages work, an awareness is developed of what information is held client side and what information is stored server side. This informs as to where vulnerabilities in web design lie, and also reinforces earlier studies about databases. | Topic based revision<br>PPQ revision |

The use of CSS Stylesheets further reinforces the desirability in general of making computer code reusable when developing solutions.

Unit 1.5.1 – Computing related legislation

**Key Skills:** Understand the purpose of and responsibility software developers have under the Data Protection Act (now the GDPR), RIPA, Communications Act and Equalities Act. Understand what Direct and Indirect Discrimination are and the responsibilities of public bodies and employers are in that respect.

**Cultural Capital:** The knowledge of the various Acts has a knock-on effect in other areas of life, particularly encouraging learners to think twice about the effects of what they may be about to post on Social Media platforms before posting it. Consideration of in direct discrimination also feed in to the user interface design of the NEA and other solutions the learner may develop as it encourages consideration for those with disabilities in such designs.

Unit 1.5.2 – Ethical, moral and cultural issues

**Key Skills:** Understand that there are a range of devices and software available to assist the disabled in using computer systems without prejudice. This follows on directly from the Equalities Act covered in the previous section. Understand the effect of computers in the workplace, their use in automated decision making, artificial intelligence, behaviour monitoring and data mining. Be able to discuss the issues of censorship that arise from the use of computers and the differing views that different cultures have towards censorship. Appreciate the environmental effect of building and using more and more devices.

**Cultural Capital:** Appreciation of disability should influence front end design of programmatic solutions. An awareness is developed of what is going on 'behind the scenes' when using services such as social media, and a 'think before you post' attitude developed. A wider understanding of cultures and governmental systems around the world is developed, along with differing viewpoints about censorship. These concepts feed directly into the study of History, Philosophy and Sociology. Consideration of what raw materials go into building computing devices (including smartphones) and the amount of energy used to power them develops an appreciation when buying a new or replacement device as to whether it is a necessity or a vanity purchase. This area has direct relevance to the study of Geography.

| | |
|---|---|
| Unit 2.1.1 – Thinking abstractly<br>**Key Skills:** Understand that abstraction is the process of filtering the necessary from the unnecessary, and developing the skill to determine which is which in a given scenario.<br>**Cultural Capital:** Abstraction is a decision tool that can be used in any problem solving or system development situation, regardless of whether the solution is computer based or not.<br><br>Ongoing: Visual Basic tutorial | |
| Second Half | |
| Unit 2.1.2 – Thinking ahead<br>**Key Skills:** Understand that thinking ahead allows situations to be anticipated and planned for, resulting in more efficient coding and the ability to write reusable code.<br>**Cultural Capital:** Thinking ahead is a decision tool that can be used in any problem solving or system development situation, regardless of whether the solution is computer based or not.<br><br>Unit 2.1.3 – Thinking procedurally<br>**Key Skills:** Understand that thinking procedurally allows for decomposition of a problem into components and subcomponents that can then be developed bottom-up.<br>**Cultural Capital:** Thinking procedurally is a decision tool that can be used in any problem solving or system development situation, regardless of whether the solution is computer based or not.<br><br>Unit 2.1.4 – Thinking logically<br>**Key Skills:** Understand that thinking logically allows for decision making in a form that can be implemented within a program, and where such decisions need to be made.<br>**Cultural Capital:** Thinking logically is a decision tool that can be used in any problem solving or system development situation, regardless of whether the solution is computer based or not.<br><br>Topic based revision<br>PPQ revision<br>End of year exam<br><br>Ongoing: Visual Basic tutorial, 'Noughts and Crosses' programming task. | Topic based revision<br>PPQ revision<br>Exam |