CURRICULUM AND ASSESSMENT PLAN

YEAR 13 | COMPUTER SCIENCE

'The curriculum and assessment of students at this stage of education has been carefully designed to promote deep learning of computer science and develop students into knowledgeable practitioners:

At KS5, students are invited to study computer science. The A'Level computing course allows students to build on the knowledge obtained from KS3 and KS4 computing. Throughout the course, students will deepen their understanding of programming using a high-level language known as Python. They will build on their understanding of the text-based language taught at key stage 3 and 4. Students will enhance their understanding with knowledge and skills in object orientated programming, recursion and 3d arrays. Much of the theory extends the knowledge gained in key stage 4 computer science with additional theory including HTML, CSS, JavaScript and databases amongst others. The computer science course is closely linked to mathematics and students will develop skills in simplifying Boolean expressions, representing decimal and negative values in binary, and widening their understanding of algorithms. The course title is OCR AS and A Level Computer Science. The course codes are: - (AS)H046, (A2) H446.

HALF TERM 1 CPU / Data Structures / Algorithms / NEA (Design & Development)

All students will know:

Component 1:
1.1 The characteristics of contemporary processors, input, output and storage devices

- 1.1.1 Structure and function of the processor, 1.1.2 Types of processor, 1.1.3 Input, output and storage (Recap on what has been covered from year 12).
- 1.1.1 (d) The use of pipelining in a processor to improve efficiency.
- 1.1.2 (b) GPUs and their uses (including those not related to graphics).

1.4 Data types, data structures and algorithms

- 1.4.1 Data types, 1.4.2 Data structures (Recap on what has been covered from year 12).
- 1.4.2 Data structures (b) Structures to store data: linked-lists, graphs (directed and undirected), stack, queue, tree, binary search tree, hash table.
- 1.4.2 Data structures (c) How to create, traverse, add data to and remove data from the data structures mentioned above.

Component 2:
2.3.1 Algorithms

- 2.3.1 Algorithms (b) The suitability of different algorithms for a given task and data set, in terms of execution time and space.
- 2.3.1 Algorithms (c) Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity).
- 2.3.1 Algorithms (f) Standard algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search).

Component 3:
Students will spend some time in lessons completing the design section of their NEA. Students will also spend time outside of lessons developing their knowledge of programming and completing the design and development sections.

All students will be assessed:

Students will complete two key assessments during this half term which will assess their understanding of each of the topics they have covered this half term. Students will also be assessed on their programming skills.
Key Assessment Point 1 – **The CPU**
Key Assessment Point 1 – **Data Structures and Algorithms**

Reading skills needed for this unit:

Students will need to be able to read a range of real-life scenarios and use decomposition.

Key vocabulary:

Pipelining, GPU, Queue, Stack, Tree, Graph, Algorithms.

HALF TERM 2 Compression / Databases / Boolean Algebra / Web / NEA (Development & Testing)

All students will know:

Component 1:
1.3 Exchanging Data ***How data is exchanged between different systems***

- 1.3.1 Compression, Encryption and Hashing, 1.3.2 Databases, 1.3.4 Web Technologies (Recap on what has been covered from year 12).
- 1.3.1 (b) Run length encoding and dictionary coding for lossless compression.
- 1.3.1 (c) Symmetric and asymmetric encryption.
- 1.3.1 (d) Different uses of hashing.
- 1.3.2 (f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy.
- 1.3.4 (b) Search engine indexing.
- 1.3.4 (c) PageRank algorithm.

1.4 Data types, data structures and algorithms

- 1.4.3 Boolean Algebra (Recap on what has been covered in year 12).
- 1.4.3 (c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.
- 1.4.3 (e) The logic associated with D type flip flops, half and full adders.

Component 2:
2.3.1 Algorithms

- 2.3.1 Algorithms (b) The suitability of different algorithms for a given task and data set, in terms of execution time and space.
- 2.3.1 Algorithms (c) Measures and methods to determine the efficiency of different algorithms, Big O notation (constant, linear, polynomial, exponential and logarithmic complexity).
- 2.3.1 Algorithms (f) Standard algorithms (bubble sort, insertion sort, merge sort, quick sort, Dijkstra's shortest path algorithm, A* algorithm, binary search and linear search).

Component 3:
Students will spend some time in lessons completing the development and testing sections of their NEA. Students will also spend time outside of lessons designing their final solution.

All students will be assessed:

Students will complete two key assessments this half term. Summative assessments will be used for each of the topics studied and will cover previous topics studied. A range of exam-style questions will be used to give students the experience needed for these topics.

Key Assessment Point 2 – **Exchanging Data**
Key Assessment Point 2 – **Algorithms and Boolean Algebra**

Exam-style questions will be used throughout lessons to prepare students to

answer questions.

Students will also be assessed during lessons through the following methods:

- Multiple choice questions
- Exam questions – past exam questions
- Live marking
- Whole class feedback
- Verbal feedback
- Programming challenges
- Self-assessment

Reading skills needed for this unit:

Students will need to be able to read a range of real-life scenarios and use decomposition. Students need to read expressions in order for them to simplify.

Key vocabulary:

Compression, Abstraction, Decomposition, Computational Thinking, Boolean Expressions, Referential Integrity, Database, SQL.

CURRICULUM AND ASSESSMENT PLAN

YEAR 13 | COMPUTER SCIENCE

HALF TERM 3 Networks / Software Development / NEA (Testing & Evaluation)

All students will know:

Component 1:
1.2 Software and Software Development
- 1.2.1 Systems software, 1.2.2 Applications generation, 1.2.3 Software development, 1.2.4 Types of programming language (Recap on what has been covered from year 12).
- 1.2.1 Systems software (h) 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.

1.3 Exchanging Data
- 1.3.3 Networks (Recap on what has been covered from year 12).
- 1.3.3 (c) Network security and threats, use of firewalls, proxies and encryption.

Component 2:
2.2 Problem solving and programming
- 2.2.1 Programming techniques, 2.2.2 Computational methods (Recap on what has been covered from year 12).

Component 3:
Students will spend some time during lesson completing their NEA for final submission. This will include carrying out vital testing and evaluating the solution they have produced to the problem they previously identified.

All students will be assessed:

Students will complete mock assessments during this period for both of the papers they will be completing (**H446/01 and H446/02**). They will also be assessed on the NEA they have produced throughout the year which is 20% of the overall grade.

Students will also be assessed during lessons through the following methods:
- Multiple choice questions
- Exam questions – past exam questions
- Live marking
- Whole class feedback
- Verbal feedback
- Programming challenges
- Self-assessment

Reading skills needed for this unit:

Students will need to read a high-level language and identify what each of the programs they read are doing. Students will need to apply inference when reading code to make predictions.

Key vocabulary:

Proxy, Virtual Machine, Firewall, Encryption, Computational methods.

HALF TERM 4 H446/01 & H446/02 Revision

All students will know:

Component 1:
1.1 The characteristics of contemporary processors, input, output and storage devices
- 1.1.1 Structure and function of the processor
- 1.1.2 Types of processor
- 1.1.3 Input, output and storage

1.2 Software and software development
- 1.2.1 Systems software
- 1.2.2 Applications generation
- 1.2.3 Software development
- 1.2.4 Types of programming language

1.3 Exchanging data
- 1.3.1 Compression, encryption and hashing
- 1.3.2 Databases
- 1.3.3 Networks
- 1.3.4 Web technologies

1.4 Data types, data structures and algorithms
- 1.4.1 Data types
- 1.4.2 Data structures
- 1.4.3 Boolean algebra

1.5 Legal, moral, cultural and ethical issues
- 1.5.1 Computing related legislation
- 1.5.2 Moral and ethical issues

Component 2:
2.1 Elements of computational thinking
- 2.1.1 Thinking abstractly
- 2.1.2 Thinking ahead
- 2.1.3 Thinking procedurally
- 2.1.4 Thinking logically
- 2.1.5 Thinking concurrently

2.2 Problem solving and programming
- 2.2.1 Programming techniques
- 2.2.2 Computational methods

2.3 Algorithms
- 2.3.1 Algorithms

All students will be assessed

Students will complete mock exams during this time. They will complete the following:
H446/01

YEAR 13 | COMPUTER SCIENCE

H446/02
The mock window is from 03.03.25 and runs until 14.03.25

Reading skills needed for this unit:

Students need to read complex text and scenarios with fluency. Students need to have a wide understanding of technical vocabulary.

Key vocabulary:
Processor, FDE Cycle, Interrupts, Procedural Languages, Methodologies, Object-Orientated, Software, Encryption, Databases, Networks, Data Structures

HALF TERM 5 H446/01 & H446/02 Revision

All students will know:

Component 1:

1.1 The characteristics of contemporary processors, input, output and storage devices
- 1.1.1 Structure and function of the processor
- 1.1.2 Types of processor
- 1.1.3 Input, output and storage

1.2 Software and software development
- 1.2.1 Systems software
- 1.2.2 Applications generation
- 1.2.3 Software development
- 1.2.4 Types of programming language

1.3 Exchanging data
- 1.3.1 Compression, encryption and hashing
- 1.3.2 Databases
- 1.3.3 Networks
- 1.3.4 Web technologies

1.4 Data types, data structures and algorithms
- 1.4.1 Data types
- 1.4.2 Data structures
- 1.4.3 Boolean algebra

1.5 Legal, moral, cultural and ethical issues
- 1.5.1 Computing related legislation
- 1.5.2 Moral and ethical issues

Component 2:

2.1 Elements of computational thinking
- 2.1.1 Thinking abstractly
- 2.1.2 Thinking ahead
- 2.1.3 Thinking procedurally
- 2.1.4 Thinking logically
- 2.1.5 Thinking concurrently

2.2 Problem solving and programming
- 2.2.1 Programming techniques
- 2.2.2 Computational methods

2.3 Algorithms
- 2.3.1 Algorithms

All students will be assessed:

Students will be assessed in the final exams they are taking during this term. They will sit two exams (**H446/01 and H446/02**).

Reading skills needed for this unit:

Students need to read complex text and scenarios with fluency. Students need to have a wide understanding of technical vocabulary.

Key vocabulary:
Processor, FDE Cycle, Interrupts, Procedural Languages, Methodologies, Object-Orientated, Software, Encryption, Databases, Networks, Data Structures

YEAR 13 | COMPUTER SCIENCE

HOW STUDENTS CAN BESUPPORTED AT HOME

Students can access a range of websites including:

Python Tutorial (w3schools.com)

HTML Tutorial (w3schools.com)

PyGame: A Primer on Game Programming in Python – Real Python

Students can also purchase OCR endorsed text books and revision guides which will allow them to develop their knowledge of the content that has been devlivered within lessons. They also have the opportunity to purchase books which consist of a range of exam-style questions.

HOW THIS LEARNING WILL BE EMBEDDED ELSEWHEREIN THE CURRICULUM

This learning can be implemented in mathematics for those that are studying this course. Students will also be able to use their knowledge of HTML and CSS to develop websites which could be used elsewhere such as in enrichment courses.