PLUME ACADEMY - LEARNING OVERVIEW

Year	11
Course	Computer Science
Specification Number/Exam Board	OCR
End of course assessment and weightings	Computer Systems: 50% Computational Thinking, Algorithms and Programming: 50%

Prior Learning

The GCSE Computer Science course builds on your child's Key Stage 3 experience by using their understanding of procedural programming techniques including sequence, selection and iteration. Key Stage 3 students also have a clear understanding of how algorithms are used in technology and how algorithms are used with data sets. Furthermore, students will have been taught about the components inside a computer and the binary number system.

Curriculum Intent – What are the curriculum aims?

GCSE Computer Science aims to develop a student's procedural programming skill so they are proficient in using programming techniques to solve problems. Programming techniques will be used in both an in class extended programming project as well as writing solutions to a given problem in the exam. Students need to be able to identify sorting and searching algorithms as well as be able to use these on a provided set of data. The course will develop the student's understanding of how computer hardware components work, which includes memory, storage, networking. This progresses on to how computing hardware is managed using an operating system and utility software. Finally, students will have explored the ethical, legal, cultural and environmental impact of technology, which includes the legislation relevant to the subject

Curriculum Implementation – What my child will be learning?

Term 1	Half Term 1	Network security Algorithms
	Half Term 2	System software Ethical, legal, cultural and environmental impacts of digital technology Boolean logic Programming languages and Integrated Development Environments
Term 2	Half Term 3	Revision and mock exams
	Half Term 4	Revision
Term 3	Half Term 5	Summer exams
	Half Term 6	

Curriculum Impact – How will progress be assessed as I learn?

Lessons are grouped into topics and each of these will have their own assessment. Assessments take the format of past exam questions. Each end of topic assessment will also contain questions from topics taught previously. Students will take a mock exam for both units, which will be in the format for a full exam.

Super-Curricular Opportunities – Support and Extending Learning

Useful study resources	If a student is really passionate about this subject...	As a parent/carer, I can assist my child in this subject by:
OCR course website page **Craig 'n' Dave revision videos** **W3Schools Python programming practice**	Follow and subscribe to various YouTube channels for both Computer Science: • Computerphile • Tech with Tim Develop your own programming projects using object-oriented programming. Visit the National Museum of Computing. Enquire about continuing to study Computer Science at A-level at Plume College.	Encouraging them to attend the after-school revision session. Purchasing our recommended revision guide (OCR GCSE 9-1 Computer Science by Collins). Encouraging them to practice programming skills at home. Support your child's development of Computer Science specific keywords and terminology.

Recall

Students could recall previous lessons in learning through starter tasks in lessons, end-of-lesson Kahoot's which usually include questions from previous parts of the curriculum, and 'no hands-up' questions by the Teacher. Additionally, students can engage in various other recall activities such as:

- **Low-stakes quizzes:** Short, frequent quizzes (paper-based or digital) that focus on recently covered material, designed more for checking understanding than for grading.

- **Think-Pair-Share:** Students first think individually about a question, then discuss their thoughts with a partner, and finally share with the class. This encourages individual recall and peer discussion.

- **Peer teaching/explaining:** Students explain a concept to a classmate, which forces them to retrieve and articulate their understanding.
- **Exam Questions:** Students regularly start lessons with exams questions form previous topics to support learning

- **Assessment:** Formal end of topic assessments will contain questions from previous learnt topics

Subject-specific terminology

1. **Two's Complement:** A method of representing signed integers (positive and negative whole numbers) in binary.
2. **Floating Point (Number):** A representation of real numbers (numbers with decimal points) in binary, allowing for a wide range of values.
3. **Character Set:** A defined list of characters (like letters, numbers, symbols), each assigned a unique binary code for computer processing.
4. **Time Complexity:** A measure of the amount of time an algorithm takes to run, typically expressed as a function of the input size.
5. **Space Complexity:** A measure of the amount of working storage (memory) an algorithm needs to execute, as a function of its input size.
6. **Hash Table:** A data structure that stores key-value pairs, using a hash function to compute an index into an array of buckets or slots for fast access.
7. **Interrupt:** A signal sent to the CPU that temporarily halts its current execution to handle a higher-priority event.
8. **Virtual Machine (VM):** A software-based emulation of a complete computer system, allowing multiple operating systems to run on one physical machine.
9. **Peer-to-Peer (P2P**): A decentralized network model where all connected devices can share resources and data directly with each other without a central server.
10. **SQL Injection:** A common web security vulnerability that allows an attacker to interfere with the queries an application makes to its database, potentially leading to unauthorized access.