

OCR Computer Science A-Level

Course Overview

- **Exam Board** – OCR
- **Usual Age Range** – 16-18
- **Qualification** – Equivalent to 1 A-Level
- **Curriculum Time** – Five 1 hour lessons per week in class plus additional work in Independent Learning Time
- **Assessment**

Component	Marks	Weight	Description
Computer Systems (01)	140	40%	The internal workings of the (CPU), data exchange, software development, data types and legal and ethical issues.
Algorithms & Programming (02)*	140	40%	Using computational thinking to solve problems.
Programming Project (03)*	70	20%	Students will be expected to analyse a problem (10 marks), and design (15 marks), develop and test (25 marks), and evaluate and document (20 marks) a program. The program must be to solve it written in a suitable programming language.

- **Grading** – A*, A, B, C, D, E, F, U
- **Full specification** - <https://www.ocr.org.uk/Images/170844-specification-accredited-a-level-gce-computer-science-h446.pdf>

Curriculum Intent

The Computer Science curriculum at our UTC equips students with the essential knowledge and practical skills to thrive in the digital age. We believe in fostering a deep understanding of computer systems, problem-solving, and computational thinking while nurturing creativity and innovation.

Beyond technical expertise, we cultivate a range of **desirable, transferable** skills like research, planning, collaboration, and effective communication of ideas. This holistic approach prepares students for diverse career paths and challenges them to think critically and solve problems creatively.

Through regular extended writing activities, we nurture a love of **reading** and hone **literacy** skills, while integrated numeracy lessons emphasize the importance of file size and compression in real-world applications.

Following this foundational program, students can seamlessly transition into further education, apprenticeships, direct employment etc. paving the way for exciting careers in the tech sector. From Graphic Design and Game Design to Animation, Software Engineering, Network Manager to Marketing and AI development, the possibilities are endless.

What can parents do to support?

Students are assessed on the basis of individual units. Students (by the end of the January sequence of lessons) are already aware of their weaknesses in certain modules of the course. To support, parents can encourage them to complete outstanding extension work (especially work in a weaker module) and take a keen interest in their child's NEA (Programming Project)

Students are also advised to practice programming, and like any other language it requires continued challenge and practice. Parents can support by encouraging students to approach: Python, Java or HTML, CSS, JS in their free time outside of class. It is recommended that students practice programming up to 3+ hours a week.

Study Tips

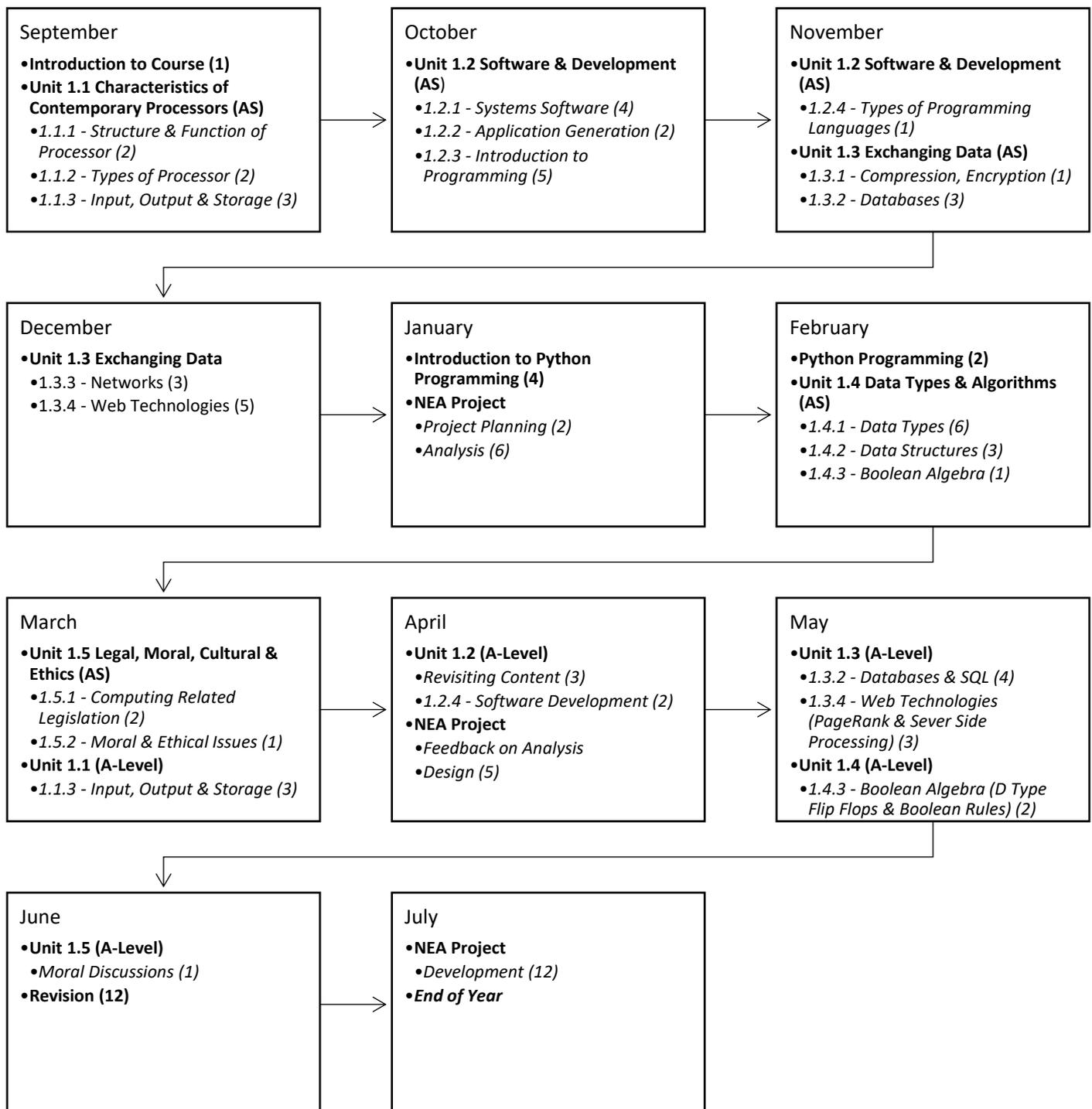
The below links will be helpful to those studying Computer Science:

- Revision Resource – <https://www.physicsandmathstutor.com/computer-science-revision/a-level-ocr/>
- Revision Resource - <https://www.savemyexams.com/a-level/computer-science/ocr/17/revision-notes/>
- IDE - <https://replit.com/~>
- Programming Resources - <https://www.w3schools.com/python/>

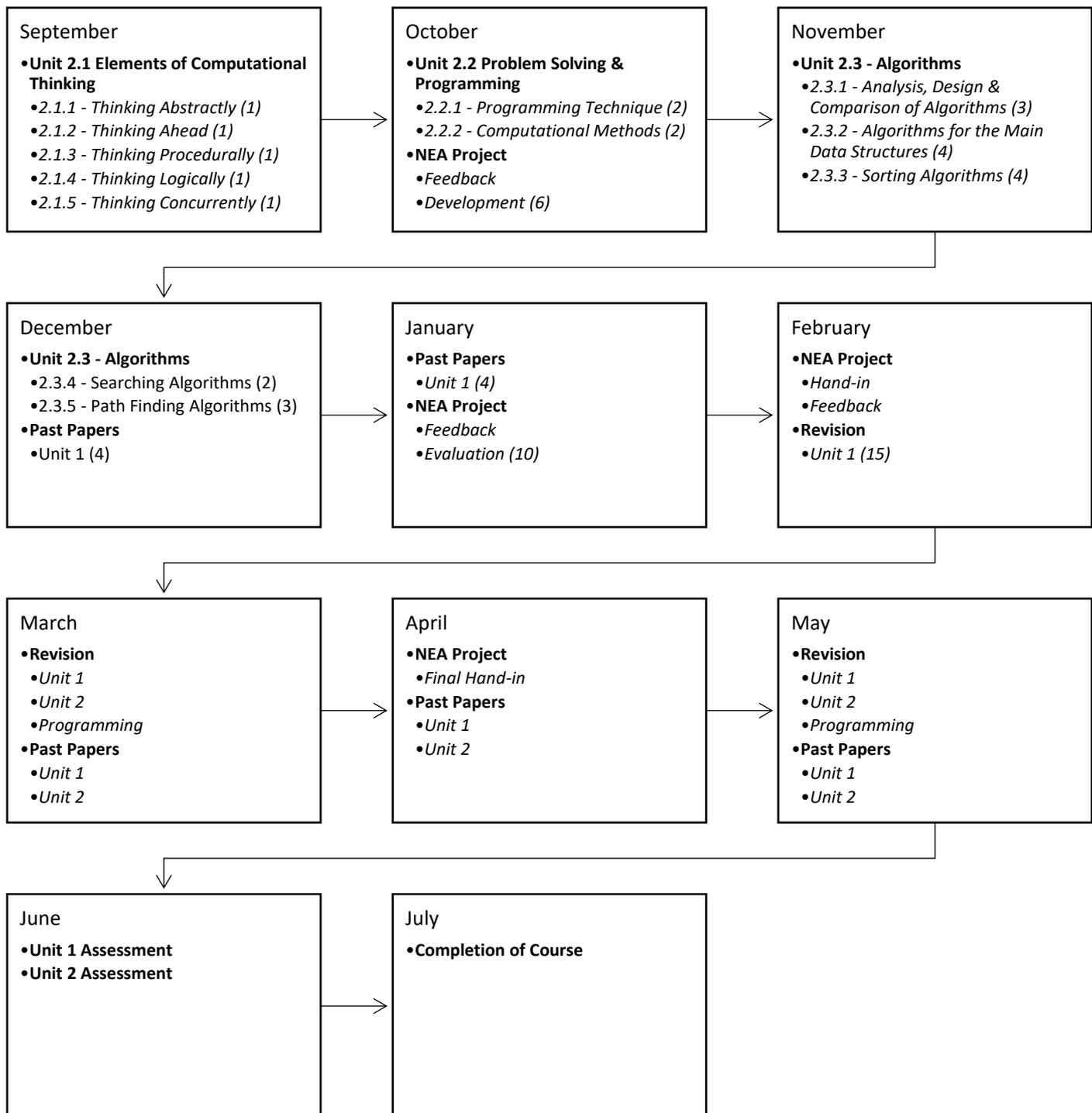
Curriculum Overview

Numbers within brackets indicate lesson time allocation ie: *1.1.1 – Structure & Function of Processor (2)* means two lessons will be allocated to the topic. Following each unit, a summative assessment will take place for gap-analysis and exam practice. Time allocated in lesson for revision

Year 12:



Year 13:



Justification of Sequence

- Unit 1's theoretical load will help to inform Unit 2's practical elements.
- Practicing Programming early will help to inform Unit 2's programming element and will support students that may have not approached programming in GCSE/ other subjects
- Unit 1's theoretical load requires students to revisit previous content to make synoptic links and a lasting long term memory
- The Programming Project's independent element requires student gain as much support in programming and ample time for feedback, questioning and answering
- Revision/ past papers will help to solidify all content and techniques acquired previously