How parents can help

Please check that your son/daughter is making full use of the resources that the department has provided:

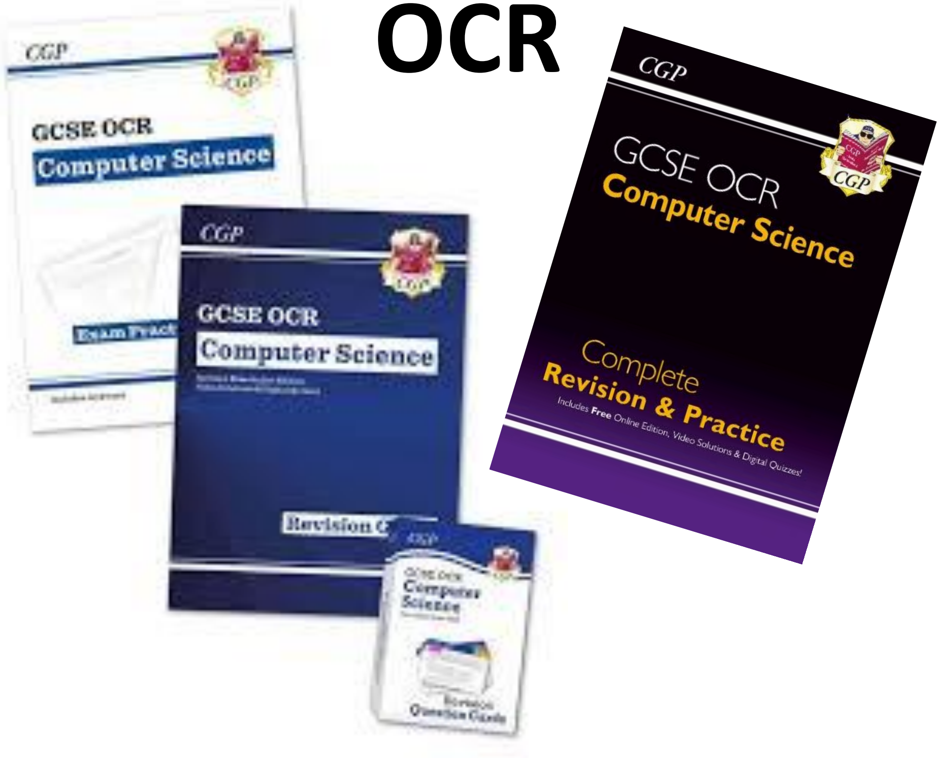
- **Teams Class Notebook:** This contains all lesson notes and primary theory. Students should review these pages regularly to consolidate learning.
- **Topic Checklists:** Ensure they have printed their revision checklists and are "ticking off" topics as they master them.
- **Visit** SmartRevise for exam questions and test subject knowledge
- **Exam Questions:** Access to a vast array of past paper questions and mark schemes, accessible via the Learning Portal.
- **Support Sessions:** Are they attending after-school sessions and making full use of lesson time for programming practice?

Encourage your son/daughter to use the most effective revision strategies:

- **Exam Practice:** Completing and marking exam questions is vital. It helps students become familiar with the command words and the level of detail required for full marks.
- **Summarising:** Encourage them to use flashcards, Mind Maps, or diagrams to break down complex topics like Networking or CPU Architecture.
- **Knowledge Retrieval:** Test them on their ability to recall key definitions from their flashcards or use online tools like **Quizlet** to store notes centrally.
- **Teach It:** Ask them to explain a key concept from their checklist to you—if they can teach it, they understand it!
- **Provide the necessary resources:**
- Flashcards, highlighters, pens, and paper.
- A quiet space with a device for revising and reviewing the Class Notebook.
- Encouragement and a reminder of what they are aiming for!

Computer Science

OCR



Advice for Effective Revision

GCSE Computer Science
What can you do to improve?

What do I need to revise?

PAPER 1: Computer Systems

Topics: •Systems Architecture •Memory and Storage •Computer Networks •Network Topologies •Protocols and Layers •Network Security •Systems Software and •Ethical, Legal, Cultural & Environmental concerns.

PAPER 2: Computational Thinking, Algorithms and Programming

Topics: •Algorithms (Search and Sort) •Programming Fundamentals •Robust Programs •Boolean Logic (Truth Tables), and •Programming Languages and IDEs

How it's assessed

Two written exams: Both are 1 hour 30 minutes.

80 marks per paper.

Each exam = 50% of the GCSE.

Where to find resources

Use the **Teams Class Notebook** to access the **Content Library**. Here you will find all the lesson resources and videos, revision checklists, your digital lesson notes, and practice papers.

Don't forget to use your Cornell Notes to revise from!

What to do

Resource	How to use it
Teams Class Notebook	Use this to access all lesson notes and primary theory. Review these pages to consolidate what was covered in class.
Topic Checklists Reflect	Use these to find the exact details needed for the spec. RAG rate your knowledge (Red, Amber, Green) to identify areas to work on.
Revision Guides & **SmartRevise** SMART REVISE Read around	Use these to help build your understanding of Paper 1 and Paper 2. Read and answer practice questions for the parts you don't understand yet.
Exam Questions - by Topic - Whole Papers Reflect	It is vital that you check you can answer exam questions on each topic. Check your answers against the mark schemes to learn the specific keywords examiners look for.
Flash Cards **and** **Mind Maps** Review	Use these to test your recall of technical vocabulary (e.g., Protocols, RAM vs ROM) and draw out system architectures.
Programming Practice (IDE) Respond to Feed Back	Use a Python IDE at home to practice writing and debugging code. You cannot learn programming just by reading about it.
Friends - studv groups	Create a study group to work through difficult logic problems or trace tables together to support and encourage each other.