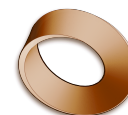


Y11 Computer Science — GCSE

Course Expectations • 2026–2027



Teacher	Mr. Alex Rawson • a.rawson@stleonards-fife.org
Room	PH1
We meet	Week A • Wednesday • Period 1 (08:55–09:50) Week A • Wednesday • Period 2 (09:55–10:45) Week B • Wednesday • Period 5 (14:05–15:00) Week B • Wednesday • Period 6 (15:05–15:55) Week B • Thursday • Period 4 (12:15–13:05)
Course page	rawsonvault.com/cs/gcse

1 What this course is

The second year of **OCR GCSE Computer Science (J277)**. Both papers are sat at the end of Year 11, there is no coursework, and nothing has been banked before then.

We finish the remaining content and then shift to proving you can do it on paper, in silence, without a computer or a calculator. That last part is worth saying plainly: you may be a good programmer and still lose marks because you have never written an algorithm by hand under time pressure. We fix that this year.

2 What we cover

Topic	Some of what is in it
Component 01 — Computer Systems	
1.1 Systems architecture	CPU, registers, fetch–execute
1.2 Memory and storage	RAM, ROM, secondary storage, compression
1.3 Computer networks	Topologies, protocols, layers
1.4 Network security	Threats, and how to defend against them
1.5 Systems software	Operating systems and utilities
1.6 Ethical, legal, cultural and environmental impacts	
Component 02 — Computational Thinking, Algorithms and Programming	
2.1 Algorithms	Searching, sorting, tracing, flowcharts
2.2 Programming fundamentals	Sequence, selection, iteration, data types
2.3 Producing robust programs	Validation, testing, defensive design
2.4 Boolean logic	Gates, truth tables, logic diagrams
2.5 Programming languages and IDEs	

All of it is examinable regardless of when we taught it.

3 What you need

- **Laptop** — charged
- **Ring binder** — the full set of theory notes, bring all of it, we go backwards constantly
- **Dividers** — for the ring binder, to keep components separate
- **Writing utensils** — pencils, pens (multiple colours preferred), a ruler, and an eraser — logic diagrams and flowcharts are drawn by hand in the exam



*A ring binder with dividers,
like this one*

No calculator in either paper, so all binary and hex arithmetic is done by hand.

4 How you are assessed

Weight	Paper	Format
50%	Component 01 — Computer Systems	80 marks, written
50%	Component 02 — Computational Thinking, Algorithms and Programming	80 marks, written

No coursework. Your grade is the two papers, and the papers are handwritten and closed-book.

If your 20-hour programming project is not finished, finish it — not for the marks, which do not exist, but because Component 02 asks you to write and trace code on paper and nothing prepares you for that like having built something real.

Code on the paper appears in OCR's Exam Reference Language. We practise reading and writing it alongside Python until neither looks strange.

5 Homework

Expect one or two pieces a week: past-paper questions, timed sections, and targeted practice on whatever the last assessment exposed.

Marked to the OCR scheme, which is fussy about wording — deliberately. Losing a mark in November for a vague definition is cheap; losing it in June is not.

6 Programming

Keep programming, even though it is not directly marked. The Component 02 questions ask you to trace, correct, and complete code, and reading code fluently is a skill that decays quickly if you stop.

Same rules as ever: save your work, keep versions, comment as you go, debug before you ask, and never hand in code you cannot explain line by line.

7 Deadlines and late work

Deadlines are given in class and posted on the course page. Work handed in late is marked and returned, but recorded as late; more than a week late is recorded as not submitted.

If something is going wrong — illness, deadlines stacking up, anything at home — **come and tell me before the deadline**. An extension asked for in advance is nearly always fine. Asking afterwards is different — by then, I usually cannot say yes.

8 Absence

Being away does not remove the work. It only delays it. **It is your job to find out what you missed** — check the course page first, then ask me.

- **Lessons.** Notes and resources are on the course page. Catch up before the next lesson.
- **Assessments.** Take the assessment within three school days of coming back. Come and book a time yourself — this is your responsibility, and I will not remind you.
- **Extended absence.** See me on your first day back and we will agree a realistic plan and a set of dates.

9 Phones, laptops, and AI

Phones stay in bags, out of sight, for the whole lesson.

Laptops are open when the task needs them and closed when it does not.

AI tools will write you working code in seconds, and this is the year that becomes genuinely dangerous:

- **Fine:** explaining an error, explaining a concept, generating extra practice questions, reviewing code you already wrote.
- **Not fine:** writing the code, the algorithm, or the six-mark answer you hand in as your thinking.
- **Always:** one line saying what you used and what for. Declaring it is never penalised. Not declaring it is.

Both papers are handwritten, closed-book, and no AI is coming into that hall with you. Every question it answers on your behalf this year is a question you will meet again in June, alone, with a pen.

10 Academic honesty

You may work together — you learn faster when you do. But **the writing must be yours, done on your own.**

- Write down who you worked with, on the work itself.
- Cite anything you used — book, website, video, diagram, dataset. Putting it in your own words still needs a citation: you are borrowing the idea even when you change the words.
- Code from anywhere — a friend, a forum, an AI — is cited, or it does not go in.

11 In the classroom

- Arrive on time and ready to start.
- Ask the question. Someone else has it too, and I would rather stop the lesson than let you fall behind.
- Wrong answers said out loud are how the lesson works. Nobody is laughed at in here.
- Leave the room ready for the next class.

12 How to do well this year

Past papers, marked honestly against the scheme, beat rereading by a distance. Do them in pen, to time, with the answers face down — then work out exactly why each lost mark was lost.

Write the theory in full sentences. Trace algorithms by hand until it is boring. Ask early: there is no version of this year where a gap you have sat on since September is easier to fix in April.

You already know most of this. This year is about turning what you know into marks.