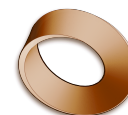


Y10 Computer Science — GCSE

Course Expectations • 2026–2027



Teacher	Mr. Alex Rawson • a.rawson@stleonards-fife.org
Room	PH1
We meet	Week A • Tuesday • Period 5 (14:05–15:00) Week A • Tuesday • Period 6 (15:05–15:55) Week A • Wednesday • Period 3 (11:15–12:10) Week B • Tuesday • Period 1 (08:55–09:50) Week B • Thursday • Period 1 (08:55–09:50)
Course page	rawsonvault.com/cs/gcse

1 What this course is

OCR GCSE Computer Science (J277), over two years, with both exams at the end of Year 11. It comes in two halves that feel like different subjects and are not: how computers actually work, and how to make them do something.

A warning worth giving on day one — this is not an ICT course. There is no coursework to carry you, the exams are written, and a good half of the marks reward precise technical writing rather than clever code. People who love programming sometimes find the theory paper a shock. Both halves are 50%.

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	

3 What you need

- **Laptop** — charged
- **Ring binder** — for theory notes. Component 01 is a written subject and needs written notes
- **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
- Your planner



*A ring binder with dividers,
like this one*

No calculator. Calculators are not allowed in the J277 exams, so we do the binary and hex arithmetic by hand from the start. It is easier than you think by about November.

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

Both papers are sat at the end of Year 11. There is **no coursework component** — your grade is the two papers.

The programming project is a required part of the course but is not marked towards your grade: roughly 20 hours building something substantial. Do not treat it as optional. Component 02 asks you to write and reason about code on paper, and the students who can do that are, without exception, the ones who have actually built things.

On the exam paper, code appears in OCR's Exam Reference Language rather than a real language. We program in Python and practise translating between the two, so neither looks strange in May.

5 Homework

Expect one homework set per week, around 1 hour — exam questions on the theory, or a programming task to finish.

Theory homework is marked to the OCR mark scheme, which is fussier about wording than you will like at first. That fussiness is the subject: "it goes faster" is not an answer, and learning why is most of the work.

6 Programming

- **Save your work where you can find it**, and keep old versions. "It worked yesterday" is not a recoverable state of affairs.
- **Comment your code** — for the person reading it in Year 11, who is you.
- **Debug before you ask:** read the error, find the line, say out loud what you expected to happen. Then ask, and I will help faster.
- **Never submit code you cannot explain line by line.** If you cannot walk me through it, it is not yours, whether it came from a friend, a forum, or an AI.

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 should stay in lockers - check the new Mobile Phone Policy.

Laptops are open when the task needs them and closed when it does not. I will say which.

AI tools matter more in this subject than any other I teach, because they will write you working code in seconds — and because you are the generation that will be paid to use them well. So the line is specific:

- **Fine:** asking it to explain an error message, explain a concept you are stuck on, generate extra practice questions, or review code you already wrote.
- **Not fine:** having it write the code, the algorithm, or the six-mark answer that you then hand in as your thinking.
- **Always:** one line at the end saying what you used and what for. Declaring it is never penalised. Not declaring it is.

The reason is entirely practical. Both papers are handwritten, closed-book, and ask you to produce and trace code on paper. Every hour an AI programmed for you is an hour you did not spend becoming someone who can do it in an exam hall — and, later, someone worth hiring to check what the AI wrote.

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 copied from anywhere — a friend, a forum, an AI — is cited like anything else, 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 in this course

Write the theory down in your own words as we go. Component 01 punishes vagueness, and you cannot revise from notes that were vague when you wrote them.

Program badly, often, and on purpose — the errors are the lesson. Do the exam questions in full sentences even when you are sure you know it; knowing it and writing it are different skills, and only one of them is marked.

Ask early. This subject stacks: 1.2 assumes 1.1, and 2.3 assumes 2.2.