

How to get a grade 5 in OCR GCSE Computer Science

A grade 5 is a strong pass. You get there by knowing the core facts of Paper 1 accurately, doing the standard calculations without a calculator, and being able to read, trace and write short programs for Paper 2. You don't need perfect answers to the hardest questions, but you do need to attempt every one.

What a grade 5 asks of you

- Recall facts and definitions precisely across 1.1 to 1.6: what each part of the CPU does, RAM compared with ROM, the types of secondary storage, network hardware, what each protocol is for, threats and how to prevent them, the functions of an operating system, and the three laws.
- Explain simple points with a reason, e.g. 'In a star topology, if one cable fails only that device loses its connection, so the rest of the network keeps working.'
- Do the standard calculations reliably: binary, denary and hexadecimal conversions, binary addition, binary shifts and file sizes.
- Read code written in OCR Exam Reference Language, complete trace tables, and write short programs with input, selection, a loop and output.
- In the extended response question, make three or four relevant points about the scenario, each with some explanation.

Secure these topics first

- 1.2 Memory and storage: start with 1.2.4 (binary, hex, characters, images and sound) and 1.2.3 (units). The calculations and definitions here are some of the most dependable marks on Paper 1.
- 1.1 Systems architecture: the jobs of the ALU, CU, cache and registers (PC, MAR, MDR, accumulator), the fetch-execute cycle, and the three factors that affect CPU performance.
- 1.3 and 1.4: LANs and WANs, network hardware, star and mesh topologies, what each protocol does, and each threat matched with a way to prevent it.
- 2.2 Programming fundamentals: variables, constants, data types, casting, selection, count-controlled and condition-controlled loops, MOD and DIV.
- 2.1 Algorithms: trace tables, and the steps of linear search, binary search and bubble sort.
- 2.3 and 2.4: the four types of test data, syntax errors compared with logic errors, and truth tables for AND, OR and NOT.

Question types to practise

- 1- and 2-mark state, identify and give questions: one exact fact per mark.
- 2-mark explain questions: a point plus a reason.
- Number conversions and file-size calculations, with full working.
- Trace tables for short loops.
- Showing each pass of a bubble sort and each step of a binary search.
- Completing a truth table and a test plan.
- Filling in the missing lines of an algorithm, and writing a short validation loop.

Revision plan, week by week

- This plan assumes about 10 weeks. If you have less time, double up the weeks but keep the mocks at the end.
- Weeks 1 and 2: 1.1 Systems architecture and 1.2 Memory and storage. Do ten minutes of conversions by hand every day.
- Week 3: 1.3 Computer networks, connections and protocols.
- Week 4: 1.4 Network security and 1.5 Systems software.
- Week 5: 1.6 Ethical, legal, cultural and environmental impacts, including writing one timed extended response answer.
- Week 6: 2.2 Programming fundamentals. Write small programs by hand, then type them in and run them.
- Week 7: 2.1 Algorithms: trace tables, searches and sorts.
- Week 8: 2.3 Producing robust programs, 2.4 Boolean logic, and 2.5 Programming languages and IDEs.
- Week 9: a timed mock of each paper, then list every subtopic where you lost marks.
- Week 10: fix those subtopics, then sit a second timed mock of each paper.
- Every week: 10 minutes of flashcards a day, and one short coding task so programming never goes cold.

How to use this website

- For each subtopic, read the notes, learn the cheatsheet, then do the worksheet and check it against the answers.
- Build a Custom paper on just that subtopic at Easy or Medium difficulty and mark it with the mark scheme.
- Open the grade-by-grade skill lists and tick off every grade 4 and grade 5 skill on the checklist as you master it.
- Do the Daily 5 every day: five questions from your weakest topics, and the streak keeps you going.
- Use Flashcards for definitions, registers, protocols and laws; spaced repetition brings back the ones you keep forgetting.
- Check Progress every week, work on your weakest subtopics first, and redo the questions you dropped marks on.
- Use the Whiteboard for trace tables, binary working and logic diagrams on screen.
- Sit the Mock exams in weeks 9 and 10, timed, one paper at a time.
- If programming isn't clicking, book the free 30-minute trial lesson with Chhetri Academy for 1-to-1 help.

Timing strategy in the exam

- Allow about a minute per mark, and don't spend five minutes on a 1-mark question.
- Paper 1: collect the quick recall and calculation marks steadily, and keep at least a minute per mark, plus planning time, for the extended response question.
- Paper 2: start Section B with at least a minute per mark left. Write something for every part: an input line and a loop can earn marks even if the program is unfinished.
- Skip anything that stalls you, mark it, and come back to it at the end.

Traps at grade 5

- One-word answers such as 'faster' or 'cheaper' with no comparison.
- Mixing up RAM and ROM, or the MAR and the MDR.
- Forgetting to divide by 8 to turn bits into bytes, or giving a file size with no unit.
- Using a binary search on an unsorted list, or showing only the final sorted list when each pass was asked for.
- Mixing up invalid data (right type, out of range) with erroneous data (wrong type).
- Leaving code questions blank.
- Writing general answers that ignore the scenario in the question.

Free notes, worksheets, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers.
Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.