

12 supercurriculars that strengthen your child's Computer Science application.

What top universities actually want to see beyond A-level grades.

the activities:

Your child doesn't need to do all 12. Pick 2-3 that genuinely interest them and go deep. Admissions tutors can spot a checklist approach immediately.

1. **Maths Olympiad (UKMT)**

Demonstrates academic depth. Oxbridge tutors genuinely love seeing this.

Start here: Enter via school or register at ukmt.org.uk. Begin with the Intermediate challenge in Year 9-11.

2. **Build and Deploy an App**

A tangible product with real users is the strongest personal statement evidence there is.

Start here: Identify a simple problem to solve. Use free tools like [React](#) and [Firebase](#). The key is deploying it so people can actually use it.

3. **Hackathons**

Shows teamwork, coding under pressure, and initiative.

Start here: Search for student hackathons on mlh.io. Many are virtual, free, and open to under-18s.

4. **Read CS Books Beyond the Syllabus**

Signals genuine passion beyond A-level content and self-directed curiosity.

Start here: "[Grokking Algorithms](#)" by Aditya Bhargava or "Code" by Charles Petzold.

the activities (continued):

5. **Start a Tech Blog**

Demonstrates communication of complex ideas, which is exactly what they'll need in interviews.

Start here: Use [Medium](#) or [GitHub Pages](#) (both free). Write about projects, concepts, or problems solved.

6. **Online Courses (Project-Backed)**

Shows self-directed learning. Must be project-backed, not just passive certificates.

Start here: [Harvard CS50 on edX](#) (free) is the gold standard. Complete the full course including the final project.

7. **Read a CS Research Paper**

Shows intellectual depth and academic curiosity at a level beyond A-level study.

Start here: Use [Google Scholar](#). Pick a topic they love (AI, cryptography, algorithms). Summarise in their own words.

8. **Open Source Contributions**

Real-world collaborative coding that admissions tutors can actually verify on [GitHub](#).

Start here: Find beginner-friendly projects tagged "good first issue" on [GitHub](#). Start small: fix a bug, improve docs, add a feature.

@EduStrategyCaroline

the activities (continued):

9. Tutor CS to Younger Students

Deepens their own understanding and shows leadership. Teaching is the best way to learn.

Start here: Run a coding club at school, volunteer with [Code Club](#), or tutor a younger student one-to-one.

10. Game Development

Creative and technical blend. Great personal statement talking point that shows range.

Start here: Use [Unity](#) (free) or [Godot](#) (free, open source). Build something simple and publish it.

11. Competitive Programming

Sharpens algorithmic thinking, which is central to CS at top universities.

Start here: [Codeforces](#) or [LeetCode](#). Start easy, work up. The [British Informatics Olympiad](#) is the UK competition.

12. University Taster Days

Shows genuine interest and gives material for the "Why this university?" interview question.

Start here: Check CS department websites from Year 11 onwards. Many run free taster days, summer schools, and public lectures.

Pick 2-3. Go deep, not wide. Quality over quantity. @EduStrategyCaroline

Need a personalised plan for your child?

Whether it's GCSE planning, A-level choices, university applications, or supercurricular strategy, I can help you build a clear plan for your child's next step.

Book a 1:1 Education Advice Call. 30 minutes. Your specific situation. A clear plan forward.

uniplan.me/shop

~~This guide shows what to do.~~ The CS Requirements Cheat Sheet shows where to apply.

Grades, offer rates, and hidden catches for 17+ universities.

£50 / 30 minutes / book this week

Follow @EduStrategyCaroline for data-driven education strategy for parents.

Find your child's actual matches (free, grade-matched) at

uniplan.me/courses