

Computer Science

Curriculum Intent: To give our A level students the opportunity to learn 'powerful knowledge' through a curriculum with computational thinking at its core. To develop our students as Computer Scientists; building the capability, ethical awareness, resilience, knowledge, and skills required to become creative problem solvers in a digital world.

Practical coding is central to our approach and students will build their skills to enable the application of computing principles such as algorithms, data representation and data structures. Students develop the skills to independently design, implement, refine, and evaluate programs.

Year 12	Year 13
<p>Core knowledge: The characteristics of contemporary processors. Input, output and storage devices. Software and software development. Exchanging data. Data types, data structures and algorithms. Legal, moral, cultural and ethical issues. Elements of computational thinking. Problem solving and programming. Algorithms to solve problems and standard algorithms.</p> <p>Procedural knowledge (how to.): Apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation. Be able to use algorithms to describe problems. Be able to analyse a problem by identifying its component parts. The capacity to think creatively, innovatively, analytically, logically and critically. The capacity to see relationships between different aspects of computer science. Be able to apply mathematical skills including Boolean logic and algebra. Use a range of programming techniques. The ability to analyse problems in computational terms through practical experience of solving such problems, including writing programs to do so the ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology. Be able to identify stakeholders in contextualised questions and write application A02 responses. The literacy skills to write technical, extended exam responses that demand evaluative A03 responses. The ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.</p> <p>Assessment: Teacher questioning in lessons. Regular review questions at the beginning of lessons to check on prior learning and challenge misconceptions Regular MCQ quizzes to check on prior learning and challenge misconceptions.</p>	<p>Core knowledge: Software and software development. Exchanging data. Data types, data structures and algorithms. Legal, moral, cultural and ethical issues. Elements of computational thinking. Problem solving and programming. Algorithms to solve problems and standard algorithms.</p> <p>Procedural knowledge (how to.): Apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation. Be able to use algorithms to describe problems Be able to analyse a problem by identifying its component parts. The capacity to think creatively, innovatively, analytically, logically and critically. The capacity to see relationships between different aspects of computer science. Be able to apply mathematical skills including Boolean logic and algebra. Be able to identify stakeholders in contextualised situations and write A02 application responses. The literacy skills to write technical, extended exam responses that demand evaluative A03 responses Use a range of project skills to analyse, design, develop, test, evaluate and document a program written in a suitable programming language. For example, write a technical specification based on user needs. To identify measurable success criteria. The literacy and software application skills to write extended technical reports. Be able to apply appropriate principles from an agile development approach to project development. The ability to articulate the individual (moral), social (ethical), legal and cultural opportunities and risks of digital technology.</p> <p>Assessment: Teacher questioning in lessons. Regular review questions at the beginning of lessons to check on prior learning and challenge misconceptions Regular MCQ quizzes to check on prior learning and challenge misconceptions.</p>

Regular exam question practice with either whole class or individual feedback.

Formal assessments which more broadly assess the curriculum including several topics in one assessment paper.

TSAT Assessments: Formal assessments which more broadly assess the curriculum including several topics in one assessment paper.

January 2024 TSAT Assessment – Class Based

Students will be assessed on topics from the Autumn Term. The assessment will be in class. A guide to the assessment will be on Satchel:One.

June 2024 TSAT Trial Exam in Exam Location

Students will be assessed on topics from both components taught in the Autumn, Spring and Summer Term. A guide to the assessment will be on Satchel:One.

Homework:

Homework is set **once per week** by each teacher.

Expect to spend up to 2 hours 30 minutes on your homework in total.

All homework tasks will be set via Satchel:One.

Typical homework will include but is not limited to:

Cornell notes made using online tutorials (pre-learning) MCQ review questions to consolidate key ideas from the A level course through Smart Revise and Isaac Computing.

Wider reading tasks to broaden your computer science knowledge.

Revision for end of topic assessments and main assessment points.

Practice exam questions to develop exam technique Learning keywords definitions and spellings which may take the form of online MCQ Quizzes or be embedded into the Cornell notes.

Links to careers and personal development include:

A-level Computer Science can open doors to various career opportunities in data science, web development, product management, engineering, software development and communications or prepare students for higher education at university. Computer Scientists often become proficient in other subjects and develop significant transferable skills. Examples of careers are discussed in lessons.

We encourage our students into summer placements and work experience with employers, hackathons and competitions.

In particular, the study of Computer Science builds the resilience of students, solving problems computationally is hard and coding is hard it requires sustained practice.

Regular exam question practice with either whole class or individual feedback.

Formal assessments which more broadly assess the curriculum including several topics in one assessment paper.

Homework:

- Homework is set **once per week** by each teacher
- Expect to spend up to 2 hours 30 minutes on your homework in total
- All homework tasks will be set via Satchel:One

Typical homework will include but is not limited to:

Cornell notes made using online tutorials (pre-learning) MCQ review questions to consolidate key ideas from the A level course through Smart Revise and Isaac Computing.

Wider reading tasks to broaden your computer science knowledge.

Revision for end of topic assessments and main assessment points.

Practice exam questions to develop exam technique.

Learning keywords definitions and spellings which may take the form of online MCQ Quizzes or be embedded into the Cornell notes.

Completing the NEA which has been started in class with short term deadlines. Deadlines will be on Satchel:One.

January 2024 TSAT Trial Exam in Exam Location

Students will be assessed on Topics from both components taught in Year 12 and the Autumn and spring of year 13. A guide to the assessment will be on Satchel:One.com

March 2024 TSAT Assessment – Class Based

Students will be assessed on Topics from Year 13 spring term. The assessment will be in class. A guide to the assessment will be on Satchel:One.

NEA Coursework:

- Programming Project.
- 20% Of final Marks.

Links to careers and personal development include:

A-level Computer Science can open doors to various career opportunities in data science, web development, product management, engineering, software development and communications or prepare students for higher education at university. Computer Scientists often become proficient in other subjects and develop significant transferable skills. Examples of careers are discussed in lessons.

	<p>We encourage our students into summer placements and work experience with employers, hackathons and competitions.</p>
--	--

In particular, the study of Computer Science builds the resilience of students, solving problems computationally is hard and coding is hard it requires sustained practice.