

Powerful Knowledge

Students follow the **OCR GCSE (9–1) Computer Science J277** specification. The course develops students' understanding of how computer systems work alongside their ability to think computationally and design, write, test and refine computer programs.

The qualification consists of two equally weighted components, each assessed by a **1 hour 30 minute written examination worth 80 marks and 50% of the final GCSE**.

Component 01: Computer Systems

This component focuses on the principles of computer systems and the theoretical knowledge that underpins modern computing. Students study:

- **1.1 Systems Architecture** – the purpose and operation of the CPU, the Fetch-Decode-Execute cycle, CPU performance and embedded systems.
- **1.2 Memory and Storage** – primary and secondary storage, units of data, binary and hexadecimal, character representation, images, sound and compression.
- **1.3 Computer Networks, Connections and Protocols** – networks, network hardware, wired and wireless connections, protocols and layers.
- **1.4 Network Security** – threats to computer systems and networks, including malware, social engineering and methods of identifying and preventing vulnerabilities.
- **1.5 Systems Software** – operating systems and utility software.
- **1.6 Ethical, Legal, Cultural and Environmental Impacts of Digital Technology** – consideration of the wider impacts of technology, including privacy, legislation, environmental issues and ethical and cultural concerns.

Component 02: Computational Thinking, Algorithms and Programming

This component develops students' ability to solve problems using computational methods and apply their knowledge through programming. Students study:

- **2.1 Algorithms** – computational thinking, designing algorithms, searching and sorting algorithms and identifying and correcting errors.
- **2.2 Programming Fundamentals** – variables, constants, operators, data types, sequence, selection, iteration, string manipulation, arrays, file handling, SQL, functions and procedures.

- **2.3 Producing Robust Programs** – defensive design, validation, authentication, maintainability and testing.
- **2.4 Boolean Logic** – logic operators, truth tables and logic diagrams.
- **2.5 Programming Languages and Integrated Development Environments** – high- and low-level languages, translators and the tools provided by an IDE.

Alongside the theoretical content, **practical programming is an essential part of the course**. Students are given regular opportunities to design, write, test and refine programs using a high-level text-based programming language. At our school, **Python** is used as the main programming language. These practical skills directly support the requirements of Component 02 and particularly the programming questions encountered in Section B of the examination.

In **Year 10**, students establish the core knowledge required for the course, beginning with Systems Architecture and Programming Fundamentals before progressing through Data Representation, Computational Thinking, Networks, System Software and the design of algorithms. Programming is revisited throughout the year so that students continually apply theoretical concepts through practical problem solving.

In **Year 11**, students build upon their programming knowledge and develop their understanding of robust and defensive programming, network security and the wider impacts of digital technology. Increasing emphasis is placed on applying knowledge to unfamiliar problems, programming examination questions and developing effective examination technique. Revision and mock examinations are used to identify and address gaps in knowledge before the final examinations.

Literacy

Computer Science has a substantial technical vocabulary and students are expected to use terminology accurately when describing systems, explaining algorithms and evaluating solutions.

Key terminology includes:

CPU · Register · Cache · Fetch-Decode-Execute Cycle · RAM · ROM · Storage · Binary · Hexadecimal · Compression · Network · Protocol · IP Address · Encryption · Malware · Authentication · Operating System · Algorithm · Abstraction · Decomposition · Sequence · Selection · Iteration · Variable · Constant · Data Type · Array · Function · Procedure · Validation · Testing · Boolean Logic · Compiler · Interpreter · IDE

Students are encouraged to communicate solutions precisely and logically. Extended-response examination questions require students to construct clear explanations and discussions using appropriate technical terminology and, where

required, consider different aspects of an issue before reaching a justified conclusion.

Programming also develops a different form of literacy: students must be able to **read, interpret, trace, write and debug algorithms and program code** accurately.

School Context

The GCSE course builds directly upon the knowledge and skills developed through our Key Stage 3 Computing curriculum.

Students enter the course having experienced computational thinking, programming, computer hardware, cybersecurity, data and the wider use of digital technology. GCSE Computer Science develops these areas in significantly greater depth and introduces the more formal terminology and problem-solving techniques required by the OCR specification.

Programming is taught as a practical skill throughout the course rather than as an isolated topic. Students regularly use Python to apply their understanding of programming constructs and to develop increasingly sophisticated solutions to problems.

Where appropriate, concepts are related to technologies students encounter in everyday life. Cybersecurity, networks, data representation, artificial intelligence, privacy and the environmental and ethical impact of technology provide opportunities to consider how Computer Science affects individuals, organisations and wider society.

Students are encouraged to become **independent computational thinkers** who can decompose unfamiliar problems, identify appropriate solutions and evaluate the effectiveness of their approach.

Assessment

Assessment at KS4 is used to monitor progress, identify misconceptions and prepare students for the style and demands of the OCR examinations.

Assessment includes:

- regular retrieval practice and low-stakes quizzes;
- practical programming activities and challenges;
- end-of-topic assessments;
- OCR-style examination questions;
- programming and algorithm-tracing questions;
- extended-response questions;

- Year 10 and Year 11 mock examinations;
- targeted revision based upon identified areas for development.

Students are increasingly exposed to past-paper and examination-style questions as the course progresses so that they become familiar with OCR terminology, command words and the level of precision required in their responses.

The final GCSE is assessed through two externally examined papers:

J277/01 – Computer Systems: 1 hour 30 minutes, 80 marks, **50% of the GCSE.**

J277/02 – Computational Thinking, Algorithms and Programming: 1 hour 30 minutes, 80 marks, **50% of the GCSE.**

Both papers are non-calculator examinations. Component 02 contains two sections, with Section B specifically requiring students to apply their programming and algorithmic knowledge.

There is **no coursework or NEA contributing to the final grade**. However, OCR requires students to be given opportunities during the course to design, write, test and refine programs, and centres confirm that this requirement has been met.

Careers

Computer Science provides students with knowledge and skills that are increasingly valuable across almost every employment sector. While the course provides an excellent foundation for further study in Computer Science and related subjects, the computational thinking and problem-solving skills developed are highly transferable.

Students develop skills in **logical thinking, problem solving, programming, analysis, decomposition, debugging, testing and evaluating solutions**. They also develop an understanding of cybersecurity, data, networks and the ethical and legal responsibilities associated with digital technology.

The course provides a foundation for further study and careers including:

- **Software Developer / Software Engineer**
- **Cybersecurity Analyst**
- **Ethical Hacker / Penetration Tester**
- **Data Analyst / Data Scientist**
- **Games Developer**
- **Web and App Developer**
- **Network Engineer**

- **Cloud Engineer**
- **Database Administrator**
- **AI / Machine Learning Engineer**
- **Computer Hardware Engineer**
- **IT Support / Systems Engineer**
- **Digital Forensics**
- **Robotics and Automation**

Computer Science skills are also increasingly important in fields such as engineering, finance, medicine, scientific research, business, manufacturing and the creative industries.