

Computer Science – Year 8

Curriculum Intent

In Year 8, Computer Science builds on the foundational understanding of systems, data and algorithms developed in Year 7. The curriculum deepens students' knowledge of how computers process data, how instructions are executed, and how problems can be solved using increasingly precise and efficient methods. Alongside theoretical understanding, students continue to develop block-based programming skills using the Scratch platform, building on foundations from Year 7 to learn key programming concepts.

What Students Will Study Across the Year

Students extend their understanding of data representation by exploring how binary is used beyond simple conversion. They learn how binary numbers can be added together, including the concept of overflow, and examine how characters are stored using binary through character sets. Students compare ASCII and Unicode and consider why different character sets are needed to represent text accurately in modern computer systems.

The curriculum develops a more detailed understanding of computer systems, focusing on the role of the CPU and how it processes instructions. Students study the main components of the CPU and learn how instructions are carried out through the fetch–decode–execute cycle. They also explore memory in greater depth, comparing primary and secondary storage and understanding the differences between volatile and non-volatile memory and their uses.

Students build on their algorithmic knowledge by studying sorting algorithms, including bubble sort and insertion sort. They learn how data can be ordered systematically and compare the efficiency of different approaches. Boolean logic is introduced through the study of logic gates, including AND, OR and NOT, enabling students to understand how decisions are made within computer systems.

Programming skills are built on using the Scratch block-based environment, developing students understanding of programming fundamentals such as inputs, outputs, variables, selection and iteration with both count and condition controlled. Students learn through scaffolded challenges with the opportunity to try, make, develop and extend programs in a visual format.

Assessment and Progression

Students complete two formal assessments across the year alongside regular classroom tasks that check understanding and application. Assessment focuses on data representation, system knowledge, algorithmic thinking and the ability to identify programming fundamentals in block-based program code. By the end of Year 8, students can explain how data is processed within a computer system and create block-based programs, preparing them well for the introduction of script-based programming in Year 9 and beyond.