



Curriculum Intent

Subject: **Computer Science**

Year: **9**

| | What? | Why? | National Curriculum Links |
|----------|---|--|---|
| Term 1-1 | <p>1.1: Systems Architecture – The CPU</p> <p>Pupils are introduced to the central processing unit (CPU) and its role as the “brain” of the computer. They study the fetch–decode–execute cycle and learn how instructions are processed. Pupils examine the main components of the CPU, including the Arithmetic Logic Unit (ALU), Control Unit (CU), cache, and registers, and how each contributes to performance. They also explore the Von Neumann architecture and how instructions and data are stored and moved within a system.</p> | <p>This unit builds directly on pupils’ understanding of data from Year 8 (<i>Data Representation</i>) by showing how that data is processed. It prepares pupils for later work on memory and storage in Year 10 (SLR 1.2) and supports programming units by giving context to how instructions are executed. This is the foundation of Paper 1 and underpins later systems knowledge.</p> | <p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> |
| | <p>Python: Input, Output, Variables, and Arithmetic Operators</p> <p>Pupils are introduced to Python syntax for basic input and output, using variables to store and manipulate values. Arithmetic operators are applied to solve simple problems.</p> | <p>These are the foundations of all programming work. This builds on Year 8 <i>Introduction to Python Programming</i> and ensures all pupils begin KS4 with a secure grasp of the basics. It prepares them for later constructs such as conditionals and loops.</p> | <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p> |
| Term 1-2 | <p>1.1: CPU Performance and Embedded Systems</p> <p>Pupils investigate the factors that influence CPU performance, including clock speed, number of cores, and cache size. They learn how these factors affect the speed and efficiency of processing. Pupils also explore embedded systems, their purposes, and examples such as washing machines, traffic lights, and mobile devices. They consider why embedded systems are designed for specific tasks and how they differ from general-purpose computers.</p> | <p>This unit builds on <i>Systems Architecture</i> by extending understanding of how the CPU works in practice. It prepares pupils for discussions of performance trade-offs in storage and networking (Year 10) and gives real-world context for the theoretical knowledge. By studying embedded systems, pupils can connect abstract CPU concepts to everyday technology. This links back to Year 7 <i>Back to the Future</i>, which introduced the history of computing, and it sets the stage for Year 10 <i>Memory and Storage</i>.</p> | <p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> |
| | <p>Python: Data Types – Integer, String, Float</p> <p>Pupils learn how different data types are represented in Python and how casting allows conversion between them.</p> | <p>Data types underpin how programs work with information. This builds on KS3 data representation knowledge and prepares pupils for comparisons, Boolean logic, and arrays.</p> | <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p> |
| Term 2-1 | <p>1.2: Data Representation: Binary, Decimal and Hexadecimal Numbers</p> <p>Pupils study the role of primary memory in a computer system, distinguishing between RAM and ROM. They examine the function of each, including how RAM stores data temporarily and ROM stores instructions permanently. Pupils also learn about virtual memory,</p> | <p>This unit builds on <i>Systems Architecture</i> and prepares pupils for further work on storage devices and secondary storage in Year 10 (SLR 1.2.2). It links back to <i>Data Representation</i> in Year 8, where pupils learned what data looks like, by showing how data is stored in memory for processing. It also prepares for later studies in GCSE Paper 1 on systems software and file management.</p> | <p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> |



| | | | |
|----------|--|--|---|
| | <p>how it extends RAM using secondary storage, and the performance issues this can cause.</p> <p>Python: Comparison Operators and Boolean Logic</p> <p>Pupils learn relational operators (<, >, ==, etc.) and Boolean operators (and, or, not). They combine these in expressions to evaluate conditions.</p> | <p>This develops problem-solving skills by enabling decision-making in programs. It links directly to Year 9 theory on Boolean logic.</p> | <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p> |
| Term 2-2 | <p>1.2: Secondary Storage</p> <p>Pupils learn the need for secondary storage to hold data and programs permanently. They study the three main types of storage: optical, magnetic, and solid state. Pupils compare the advantages and disadvantages of each in terms of capacity, speed, portability, durability, reliability, and cost. They also learn to select suitable storage devices for different applications.</p> | <p>This unit extends pupils' knowledge of memory from the previous half term, showing how long-term data is stored and retrieved. It builds on the decision-making skills developed in Year 7 (<i>Modelling Data</i>) and prepares for Paper 1 topics in Year 10 that examine storage trade-offs in more detail. It also links forward to later content on systems software, where file management and storage types are key.</p> | <p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> |
| | <p>Python: IF, ELIF, ELSE statements</p> <p>Pupils use selection statements to control program flow based on conditions. Nested if-statements are introduced..</p> | <p>Builds on Boolean expressions to enable more sophisticated solutions. It prepares pupils for handling input validation and robust program design later in KS4.</p> | <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p> |
| Term 3-1 | <p>1.2: Units and Data Capacity</p> <p>Pupils study the units of data storage, including bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, and petabyte. They learn how to calculate file sizes and storage requirements. Pupils also explore the binary representation of data and why all data must be converted into binary for processing.</p> | <p>This unit reinforces the binary knowledge from Year 8 (<i>Data Representation</i>) and connects it to storage concepts from Term 2. It prepares pupils for more advanced calculations and problem-solving required at GCSE Paper 1 and links directly to compression and data handling in Year 10. It also develops transferable maths skills used across science and technology subjects.</p> | <p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> |
| | <p>Python: FOR loops</p> <p>Pupils use for loops to iterate through ranges, lists, and strings.</p> | <p>Builds on sequence and selection by introducing iteration. Links forward to arrays and file handling where loops are essential.</p> | <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p> |
| Term 3-2 | <p>1.2: Data Representation</p> <p>Pupils consolidate their understanding of binary and extend it by practising conversions between binary, denary, and hexadecimal. They also learn binary addition and how overflow errors can occur. Pupils are introduced to how hexadecimal simplifies binary representation for humans. This unit serves as both revision of prior knowledge and extension into new areas.</p> | <p>This unit revisits and strengthens learning from Year 8 (<i>Data Representation</i>), ensuring pupils are confident with core binary concepts before moving on. It prepares pupils for more advanced binary manipulation and representation of text, images, and sound, which they will cover in Year 10 (SLR 1.2.4). This creates a smooth progression into Paper 1 topics and reinforces problem-solving with number bases.</p> | <p>4.1: develop their capability, creativity and knowledge in computer science, digital media and information technology</p> |
| | <p>Python: WHILE loops</p> <p>Pupils use while loops for repetition based on a condition, including input validation problems.</p> | <p>Reinforces defensive programming and introduces infinite loop dangers. Links to robust programming and algorithm design in Year 10.</p> | <p>4.2: develop and apply their analytic, problem-solving, design, and computational thinking skills</p> |

Chesterton Community Sports College

