

Oasis Sholing Computing: Vision



At Oasis Sholing we have a shared vision to:

- 1) Build a deep understanding of computer science, and the ability to solve problems, to enable our students to use computers to overcome barriers.**

At Sholing we challenge the erroneous preconception that young people are automatically ‘digital-natives’. We stretch and encourage our students to understand the principles of computing, and to develop their problem-solving abilities. Generative AI presents new challenges to the computer science workforce; our students will need to be imaginative and adaptable to be successful in a world we cannot accurately predict.

- 2) Develop our students’ ethical character and digital competence to enable them to positively contribute to our global community.**

We share our passion for the ethical issues presented by all forms of computing in the world around us. Our students gain a strong conviction of the need for computer scientists to be moral digital citizens, of the role computer scientists must play for computers to benefit our global society and environment, and to challenge new threats to humanity and our precarious planet.

- 3) Ensure outstanding outcomes for all, tackling social injustice by raising the inclusivity of computer science, with a focus on SEND and societal bias.**

Gender imbalance is significantly higher in the new subject of computing. There are disproportionately few girls in computer science. In 2017, the Royal Society reported that gender balance was ‘the most significant diversity issue’ in the subject. At Sholing, we have a passion to ignite passion for Computing for all students. Further, SEND students are faced with a steep learning curve when accessing computers, which we are committed to overcoming with scaffolding, regular feedback, and personalised support.

Our Curriculum

We have structured our curriculum around the most significant concepts in computing and mapped students' development of these concepts through lesson sequences and topics.

We began by going back to first principles, independently formulating three broad concepts.

This was not unlike OFSTED's own 2022 Research Review, which synthesized the following Principles of Progression; computer science, information technology, and digital literacy.

Sholing's Pillars of Computing are semantically similar in part, but with key differences such that they are our own.

Sholing's Three Pillars of Computing:



To give our curriculum a deeper structure, when designing it, we have thought about how students' knowledge of those concepts enables them to answer our "big questions" in greater depth. We aim for all students to leave Year 11 being able to meaningfully answer each using their knowledge of our core concepts (often called 'big ideas'). This enables our teachers to think about both deepening students' knowledge of concepts and developing meaningful connections between those key concepts. Our big questions are written so they make sense to students at all stages of their secondary computing curriculum.

Computation	Solution	Application
<p>How do we represent problems in their simplest form?</p> <p>Core Concept: Abstraction</p>	<p>Is our solution easy for users to operate?</p> <p>Core Concept: Usability</p>	<p>How have computers brought about the Information Age?</p> <p>Core Concept: Communication</p>
<p>How do computers make decisions?</p> <p>Core Concept: Logic</p>	<p>Is every part of our solution independently useful and interchangeable?</p> <p>Core Concept: Modularity</p>	<p>How do we control access to personal information?</p> <p>Core Concept: Privacy</p>
<p>How can we solve big problems with simple instructions?</p> <p>Core Concept: Decomposition</p>	<p>How do we ensure our solutions can be used by other computer scientists?</p> <p>Core Concept: Readability</p>	<p>How can people benefit from abusing computers?</p> <p>Core Concept: Exploitation</p>
<p>How do computers solve problems?</p> <p>Core Concept: Algorithm</p>	<p>How can solutions be broadly functional on all similar problems?</p> <p>Core Concept: Generalisation</p>	<p>How can we protect computers and people from exploitation?</p> <p>Core Concept: Security</p>
	<p>How do we solve a problem with minimum steps and space?</p> <p>Core Concept: Efficiency</p>	<p>How can computers save humans time?</p> <p>Core Concept: Automation</p>
	<p>How do we design solutions to cope with increasing amounts of data or users?</p> <p>Core Concept: Scalability</p>	<p>How can information technology achieve the goals of institutions and companies?</p> <p>Core Concept: Facilitation</p>

Sholing's Three Pillars of Computing:



- **Computation**

Computation is the process of representing, processing, and manipulating data to solve problems, forming the foundation for understanding how computers work and how they can be used to address real-world challenges efficiently. It involves key concepts such as abstraction to simplify problems, algorithms to outline step-by-step solutions, logic to guide decision-making, and decomposition to break down complex tasks. Computation also explores the underlying mechanisms of technology - from the binary operations of wires and switches in a CPU to programming structures, data storage methods like magnetic and flash memory, and how all forms of media are represented digitally. Ultimately, computation helps demystify modern computer systems, allowing us to understand how they function rather than view them as magical.

- **Solution**

A solution is a purposeful design created to address a specific problem through computation. It goes beyond simply solving the problem - it must be efficient, usable, and adaptable to similar challenges. A well-crafted solution considers modularity, readability, and scalability, making it easy for others to understand, improve, and re-use. Generalisation ensures the solution can be applied in broader contexts, while usability keeps human interaction at the core. Ultimately, the effectiveness of computation depends on the quality and thoughtfulness of the solutions we design.

- **Application**

Application refers to the real-world use of computing knowledge to address societal and environmental challenges. It involves understanding and engaging with key principles such as communication, privacy, automation, and security, while also recognizing the widespread cultural impact of machines that now store our information, drive automation, and learn in ways beyond human comprehension. Application encourages critical and creative engagement with technology, aiming to ensure that its growing influence is managed responsibly and used to contribute positively to the global community.

Year 10 Long term plan



Brief overview

Students will gain a comprehensive understanding of key computing concepts, beginning with Boolean logic and its applications in truth tables, logic circuits, and digital decision-making. They will explore computer architecture, focusing on the Von Neumann model, key components such as RAM, HDDs, and SSDs, and comparisons across speed, cost and capacity. System software fundamentals, including firmware, OS functions, and user interface management, will be taught alongside peripheral handling and device drivers. Networking knowledge will span local to global systems, addressing topologies, protocols, and data transmission. Learners will also develop strong computational thinking through abstraction, decomposition, and algorithms. Embedded systems like microcontrollers will be contrasted with general-purpose computers, using practical examples. Binary and hexadecimal operations, overflow errors, two's complement, and data encoding for text, images, and sound will reinforce digital representation. Students will apply this knowledge through programming, using correct sequence, syntax, logical operators, loops, subprograms, and input validation. Emphasis will be placed on modular code design, data structures like lists and two-dimensional arrays, and text/CSV file handling. By the end of the term, students will be equipped with the theoretical and practical skills needed for effective problem solving, digital communication, and program development.

Software Packages: Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition, they will use Python, Mu IDE

Unit	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Unit title	Compute	Interface	Systems	Information	Encode	Media
Core Concepts	<ul style="list-style-type: none"> Logic Decomposition Usability Algorithm 	<ul style="list-style-type: none"> Abstraction Algorithm Generalisation 	<ul style="list-style-type: none"> Automation Scalability Modularity Communication 	<ul style="list-style-type: none"> Communication Abstraction Scalability 	<ul style="list-style-type: none"> Abstraction Readability 	<ul style="list-style-type: none"> Abstraction Facilitation
Declarative Knowledge	<p>Students will understand Boolean logic, including AND/OR/NOT operations. They will create and interpret truth tables, build logical expressions and logic circuits, translating between them. They will also design logic diagrams and understand how these underpin digital systems and computing decisions.</p> <p>Students will understand the architecture of computer systems, including Von Neumann architecture where the CPU, memory and instructions interact through the FDE cycle. They will identify components like the RAM, which holds temporary data, and storage devices such as HDDs and SSDs, comparing them in terms of speed, durability, capacity and cost. They will also learn how storage is measured to gain a solid grounding in computing terminology and internal operations.</p> <p>Students will explore system software, including the nature of firmware, operating systems, and how Operating Systems handle memory and CPU management.</p>	<p>Students will study file systems, users, and peripheral communication. They will also see how user interfaces allow users to accessibly navigate computer systems.</p> <p>Students will be able to distinguish between input and output devices, understanding their functions and how users interact with computers. They will learn about device drivers and how they communicate with hardware, gaining insight into how systems handle peripheral devices.</p> <p>By the end of this unit, students will develop skills in computational thinking such as abstraction (removing unnecessary detail), decomposition (breaking problems into smaller parts), pattern recognition, and algorithmic thinking. They will understand how these apply in both real-life and digital contexts, providing strong problem-solving techniques.</p>	<p>Students will explore embedded systems such as microcontrollers in appliances and smart devices, and how they differ from general-purpose computers in terms of power and function. They will recognise examples like washing machines or traffic lights and understand their constraints.</p> <p>Students will understand what a network is and how devices are connected, exploring scaling from LANs to WANs such as the internet. They will understand wired and wireless technologies, and network hardware including routers and switches.</p>	<p>They will also develop an understanding of network topologies, and how data is transmitted across a network through packets. Students will also explore the importance of network protocols, and how data travels through various internet layers. By the end of this unit, students will also understand domain name systems, addressing, and how networks are vital for modern communication.</p> <p>Students will understand how data is represented using binary numbers, converting between binary and denary interchangeably, and perform binary addition on 8-bit strings.</p>	<p>They will understand the nature of overflow errors, the use of two's complement for negative numbers, and the importance of binary shifting. They will use this to further analyse the ranges of values using the expression 2^n. Following this, they will learn to use hexadecimal and convert between this and binary and practice these operations for fluency.</p> <p>Students will explore the evolution of information encoding, learning how computers encode text using ASCII and Unicode.</p>	<p>Students will study the properties of digital images including bitmaps, colour depth, resolution, file size, and digital sound including sample rate and bit depth. They will learn how to calculate storage needs, and analyse the nature of magnetic, optical and solid-state storage, understanding their characteristics, benefits, drawbacks and read/write processes.</p> <p><i>There is less declarative knowledge in this half term due to the End of Year 10 Assessments and Work Experience.</i></p>
Procedural Knowledge	<p>Students will become fluent in writing programs with correct sequence, understanding how interpreters translate code into machine code that can be understood. They will learn how to handle syntax errors, declare variables, accept user input (including casting types), apply arithmetic, and effectively use constants when necessary.</p>	<p>Students will build on their understanding of Boolean logic, using IF-ELSE and ELIF statements, and use nested conditions. They will also learn to combine conditions in python with logical operators and understand the importance of input validation to improve the reliability of their programs.</p>	<p>Students will be able to enhance their programs with while and for loops, control infinite loops, and implement exit conditions. They will use nested loops and create pattern-based programs that demonstrate strong iteration control in their coding.</p>	<p>Students will understand how to use and define both types of subprograms (procedures and functions), pass parameters, return values, and differentiate between local and global variables. They will plan and break code down into reusable chunks, emphasising modularity and efficiency.</p>	<p>Students will work extensively with lists to create, append, modify and iterate, and study string manipulation.</p>	<p>Students will learn character-based functions and practice validation, reformatting, and the reading and writing of text documents and CSV files.</p> <p>Further, students will understand two-dimensional lists and their implementation, and the nature of game save files.</p>

Year 10 breakdown by lesson



Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Compute	Interface	Systems	Information	Encode	Media
Theory					
B1.1 Boolean Operators B1.2 Truth Tables B1.3 Complex expressions B1.4 Logic Diagrams B1.5 Designing logic circuits H1.1 Hardware - An Introduction H1.2 Architecture H1.3 FDE H1.4 RAM H1.5 Storage H1.6 Comparing Storage S1.1 Firmware S1.2 Operating Systems S1.3 CPU and Memory	S1.4 Files and Users S1.5 User Interface H2.1 Inputs H2.2 Outputs H2.3 Drivers & Application A1.1 Abstraction A1.2 Decomposition A1.3 Algorithms A1.4 Pattern Recognition A1.5 Applications	H3.1 Definition and examples H3.2 Microcontrollers H3.3 Differences from General purpose H4.1 What is a network? H4.2 Network scaling H4.3 Wired + Wireless H4.4 Network Hardware	H4.5 Topologies H4.6 The Internet H4.7 Packets H4.8 Protocols 1 H4.9 Layers H4.10 Protocols B2.1 Binary Numbers B2.2 Binary to Denary B2.3 Denary to binary B2.4 Addition	B2.5 Overflow B2.6 Negatives B2.7 Value ranges (2 ^N) B2.8 Shifts B2.9 Binary Practice B2.10 Hexadecimal B2.11 Hex conversion practice B3.1 History of recorded information B3.2 Encoding Text B3.3 ASCII conversions B3.4 Bitmaps	B3.5 Colour Depth, Res, and Image size B3.6 Sampling B3.7 Bit depth and Sample rate B3.8 Graphing sound B3.9 Units of Storage B3.10 Calculating File Sizes B3.11 R/W Magnetic Storage B3.12 R/W Optical Storage B3.13 R/W Solid State storage
Programming					
P1.1 Instructing Computers P1.2 Output & Syntax Errors P1.3 Variables & Assignment P1.4 Inputs and Casting P1.5 Arithmetic P1.6 Constants P2.1 Boolean logic and comparisons P2.2 If statements, relational ops	P2.3 ELSE and ELIF P2.4 Nested Conditions P2.5 Combining Conditions P2.6 Input Validation P3.1 WHILE (loops) P3.2 Infinite loops and Escape Conditions	P3.3 FOR (loops) P3.4 Counting down P3.5 Nested loops and patterns P4.1 Calling Subprograms, built-in	P4.2 Defining procedures P4.3 Parameters & Generalisation P4.4 Functions & Return P4.5 Scope P4.6 Planning in subprograms	P5.1 Lists P5.2 Modifying Lists P5.3 Iteration P5.4 Ord and Chr P5.5 Checking Strings P5.6 Changing Strings	P5.7 Reformatting + Validation P5.8 Reading Text P5.9 Writing text P5.10 Saved games

Year 11 Long term plan



Brief overview

Overview: Autumn 1 starts with some recapping of the most common programming techniques that are used in almost all Python programs. The first lesson back will cover some of the very basics as many students will not have created a program over the summer. The complexity of programs will build very quickly so that in Spring 1 they can focus on independently building solutions to a variety of problems like those they will encounter in the assessment.

Theory lessons will start looking at some specific algorithms for searching and sorting. Students will be able to recognize and compare these algorithms allowing them to choose the best one for a given situation and justify their choice. In Autumn 2 they will look at computer networks and Spring 1 will focus on the issues and impact of computing. This unit will cover some of the same material as the cybersecurity unit in year 10, however, now students will be focusing more on the impact of cybercrimes when they occur.

Software Packages: Students will use a combination of a web browser, Formative.com, MS Teams and MS OneNote throughout each unit. In addition, they will use Python, Mu IDE.

Unit	Autumn 1	Autumn 2	Spring 1
Unit title	Nesting and Algorithms	Data structures + Networks	Program design + Ethics
Core Principles	<ul style="list-style-type: none"> Use a wide range of software and technology Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons 	<ul style="list-style-type: none"> Use a wide range of software and technology Use computational thinking skills to solve real world problems Understand what networks are and how they are used Understand simple Boolean logic Maximising the use of Horizons 	<ul style="list-style-type: none"> Use a wide range of software and technology Be aware of the risks of technology and how they can be minimised Use computational thinking skills to solve real world problems Understand simple Boolean logic Maximising the use of Horizons
Declarative Knowledge	<ul style="list-style-type: none"> Recap programming Know the purpose and process of linear and binary search, and merge, insertion, and bubble sort Know the advantages of different sort algorithms Know how to measure and improve algorithms efficiency 	<ul style="list-style-type: none"> Know the structure and purpose of csv files Know the syntax for interacting with csv files in python Recap python data structures and file handling Know the advantages and disadvantages of wired and wireless networks Know the process by which networks are accessed Know the different pieces of hardware used for networks Know the TCP/IP model and layers of the Internet Know the difference between client server and peer to peer networks 	<ul style="list-style-type: none"> Know the 3 laws relevant to computing, Data Protection, Computer Misuse, Copyright Know the different ethical legal cultural environmental privacy issues surrounding technology Know how technology has and continues to impact society, including knowledge of some emerging technologies Know some of the penalties for committing cybercrimes
Procedural Knowledge	<ul style="list-style-type: none"> Recap Programming How to apply search and sort algorithms, showing the steps How to write functions for each of the search and sort algorithms 	<ul style="list-style-type: none"> Recap python data structures and file handling How to write programs that read and write from csv files How to debug file handling errors How to suggest a type of network and for a given purpose 	<ul style="list-style-type: none"> How to design and write a program to a given brief How to discuss relevant ethical legal cultural environmental issues to a given scenario

Year 11 breakdown by lesson



Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1
Nesting and Algorithms	Data structures + Networks	Program design + Ethics	Revision	Revision
<p>Practical</p> <ol style="list-style-type: none"> Sequence Selection Iteration Input and debugging Subroutines String Handling Data Validation <p>Theory</p> <ol style="list-style-type: none"> Linear Search Binary Search Comparing searching Algorithms Coding searching algorithms Bubble Sort Merge Sort Comparing sort algorithms Coding sorting algorithms Fit for purpose? Improving algorithms Assessment 	<p>Practical</p> <ol style="list-style-type: none"> Arrays and lists 2D arrays and lists Structured programming Working with text files Working with CSV files Good programming practice Debugging <p>Theory</p> <ol style="list-style-type: none"> What is a computer network The client server Model Network Hardware Network topologies Wired and wireless transmission media Network performance What is the internet Hosting services Protocols The TCP/IP Model Assessment 	<p>Practical</p> <ol style="list-style-type: none"> Creating programs to solve problems 1 Creating programs to solve problems 2 Creating programs to solve problems 3 Creating programs to solve problems 4 Creating programs to solve problems 5 Creating programs to solve problems 6 <p>Theory</p> <ol style="list-style-type: none"> The impact of technology The law, data protection The law, Copyright The Computer Misuse Act Cultural Impacts Privacy and Surveillance Environmental impact Ethical Impact Cybercrimes Defending against cybercrimes Consequences of cybercrimes Assessment 	<p>Our revision sequence is written in response to the emerging needs of our GCSE class and the individuals within.</p> <p>Revision lessons are planned and delivered to maximise student progress by supporting students to demonstrate their declarative and procedural knowledge in the P1 and P2 terminal GCSE exams and will draw upon any content from year 10 and year 11.</p>	