

1 s t  F o r m	Term	1	2	3	4	5	6
	Title	Introduction to Computing	Why Computing?	Can a Computer Think for Itself?	Can a Computer Think for Itself? II	What Links us All?	What Links us All? II
Prior Knowledge		<ul style="list-style-type: none"> <li>How to use applications to complete classwork</li> </ul>	<ul style="list-style-type: none"> <li>What an algorithm is</li> <li>What the four computational thinking skills are</li> </ul>	<ul style="list-style-type: none"> <li>Understanding of the three programming constructs</li> </ul>	<ul style="list-style-type: none"> <li>Hardware of a computer</li> </ul>	<ul style="list-style-type: none"> <li>What digital threats exist and solutions to these</li> <li>Networking hardware</li> </ul>	
Core Knowledge	<ul style="list-style-type: none"> <li>Understand how to use main Google applications for classwork for all subject (Includes Docs, Sheets, Slides, G-Drive)</li> <li>Importance of digital footprints</li> </ul>	<ul style="list-style-type: none"> <li>The difference between hardware and software</li> <li>What algorithms are and how these can be visually represented</li> <li>The four elements of computational thinking</li> <li>E-safety</li> <li>Technology in modern society</li> </ul>	<ul style="list-style-type: none"> <li>Programs are only as effective as their algorithms</li> <li>Understanding the three core constructs of programming</li> <li>How to use a block-based programming language</li> <li>What variables are and how they are used in computer science</li> </ul>	<ul style="list-style-type: none"> <li>What logic gates are</li> <li>Implementing variables into programs</li> <li>Simple Boolean logic</li> <li>Searching algorithms and comparing their efficiencies</li> </ul>	<ul style="list-style-type: none"> <li>What the internet and world wide web are</li> <li>Different types of networks and relevant hardware for accessing the internet</li> <li>Networking hazards and ethics</li> <li>Use of creative software to create digital graphics</li> </ul>	<ul style="list-style-type: none"> <li>Using HTML for pupils to create their own websites</li> <li>Use of CSS</li> <li>Understanding how networking threats can be neutralised</li> </ul>	
Key takeaways for future learning	<ul style="list-style-type: none"> <li>Students will have basic digital literacy skills to aid their school life</li> </ul>	<ul style="list-style-type: none"> <li>An algorithm is a series of instructions to solve a problem</li> <li>Computational thinking has four elements</li> <li>Technology in modern society</li> </ul>	<ul style="list-style-type: none"> <li>Sequence, selection and iteration are the three programming constructs</li> <li>Use of block-based programming language</li> </ul>	<ul style="list-style-type: none"> <li>A variable is a piece of temporary data that can change</li> <li>Computers respond to inputs with an output</li> <li>Logic gates and their sequencing</li> </ul>	<ul style="list-style-type: none"> <li>There are many networking security threats such as phishing and shouldering</li> <li>How to create your own digital assets</li> </ul>	<ul style="list-style-type: none"> <li>How websites are created using HTML and CSS</li> </ul>	
2 n d  F o r m	Term	1	2	3	4	5	6
	Title	What's Behind the Block?	What's Behind the Block? II	Does Technology Rule?	Does Technology Rule? II	Where Can Computing Take Me?	Where Can Computing Take Me?
Prior Knowledge	<ul style="list-style-type: none"> <li>Understanding of the three programming constructs</li> <li>Know what a variable is</li> <li>What the four computational thinking skills are</li> </ul>	<ul style="list-style-type: none"> <li>How to apply the three programming constructs into a written programming language (Python)</li> </ul>	<ul style="list-style-type: none"> <li>What an algorithm is</li> <li>Computers respond to inputs with an output</li> <li>Representing algorithms as flowcharts</li> </ul>	<ul style="list-style-type: none"> <li>Computers respond to inputs with an output</li> <li>Use of block-based programming language</li> </ul>	<ul style="list-style-type: none"> <li>Technology in modern society</li> <li>Computing in the working world</li> <li>Understanding of HTML and CSS</li> </ul>	<ul style="list-style-type: none"> <li>Technology in modern society</li> <li>Computing in the working world</li> <li>Understanding of HTML and CSS</li> </ul>	

	<b>Core Knowledge</b>	<ul style="list-style-type: none"> <li>Understand how to output text to the monitor using Python</li> <li>Use of sequence, selection and iteration on Python</li> <li>Using computational thinking to solve computational problems</li> </ul>	<ul style="list-style-type: none"> <li>Understanding Python conventions</li> <li>Difference between a logic and syntax error</li> <li>Applying Python knowledge for pupils to create their own creative gamebook</li> <li>Understanding the difference between variables and arrays</li> </ul>	<ul style="list-style-type: none"> <li>Flowcharts can represent the logic of control technology</li> <li>Subroutines and their uses</li> <li>Purposes of different computer hardware (CPU, RAM, ROM)</li> <li>Ethics relating to Computer Science</li> </ul>	<ul style="list-style-type: none"> <li>Plan their own control technology system</li> <li>Use Scratch to create their own working control system</li> <li>How computers understand and represent different piece of data including binary maths</li> <li>Future control technologies and how these will affect society</li> </ul>	<ul style="list-style-type: none"> <li>How modern technology has impacted modern society, and its effects on 21<sup>st</sup> century workplace</li> <li>Undertaking creative projects using modern software (Image manipulating, creating websites, animations, cybersecurity)</li> </ul>	<ul style="list-style-type: none"> <li>Undertaking creative projects using modern software (Image manipulating, creating websites, animations, cybersecurity)</li> </ul>
	<b>Key takeaways for future learning</b>	<ul style="list-style-type: none"> <li>How to apply the three programming constructs into a written programming language (Python)</li> </ul>	<ul style="list-style-type: none"> <li>How to apply the three programming constructs into a written programming language (Python)</li> <li>The difference between variables and arrays</li> <li>Representing algorithms as flowcharts</li> </ul>	<ul style="list-style-type: none"> <li>Ethical issues relating to CS such as self-driving cars and automation in the workforce</li> <li>Computing hardware relating to CPU, RAM &amp; ROM</li> <li>Subroutines improve the efficiency of algorithms</li> </ul>	<ul style="list-style-type: none"> <li>Algorithms that complete the same task can be superior due to their efficiency</li> <li>How computers represent different data, and understand inputs</li> <li>Binary addition using nibbles</li> </ul>	<ul style="list-style-type: none"> <li>How to use a magnitude of different software to create creative projects</li> <li>Understanding of what skills are necessary in a modern economy</li> </ul>	<ul style="list-style-type: none"> <li>How to use a magnitude of different software to create creative projects</li> <li>Understanding of what skills are necessary in a modern economy</li> </ul>
<b>3</b>	<b>Term</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>r</b>	<b>Title</b>	Algorithmic Thinking	Data Representation	Python Explained I	Python Explained II	Hardware & software	Hardware & software II
<b>d</b>							
<b>F</b>	<b>Prior Knowledge</b>	<ul style="list-style-type: none"> <li>What an algorithm is</li> <li>Algorithms can be represented as a flowchart</li> <li>The four computational thinking skills</li> </ul>	<ul style="list-style-type: none"> <li>How computers represent different data, and understand inputs</li> <li>Logic gates and their sequencing</li> <li>Binary addition using nibbles</li> </ul>	<ul style="list-style-type: none"> <li>How to apply the three programming constructs into a written programming language (Python)</li> <li>Subroutines improve the efficiency of algorithms</li> </ul>	<ul style="list-style-type: none"> <li>Writing pseudocode using correct conventions</li> <li>Breaking computational problems into inputs, processes and outputs</li> <li>Use of all three programming constructs</li> </ul>	<ul style="list-style-type: none"> <li>Computing hardware relating to CPU, RAM &amp; ROM</li> <li>Computers respond to inputs with an output</li> </ul>	<ul style="list-style-type: none"> <li>Computing hardware relating to CPU, RAM &amp; ROM</li> <li>Ethical issues relating to CS such as self-driving cars and automation in the workforce</li> </ul>
<b>o</b>							
<b>r</b>	<b>Core Knowledge</b>	<ul style="list-style-type: none"> <li>Apply computational thinking skills to real-world and computational problems</li> <li>Representing algorithms as flowcharts and pseudocode accurately</li> </ul>	<ul style="list-style-type: none"> <li>Understanding the binary and denary system, and conversion between them</li> <li>Manipulating binary numbers</li> <li>Using logic gates to represent boolean logic</li> </ul>	<ul style="list-style-type: none"> <li>Use of all programming constructs within Python</li> <li>Use of lists within programs and their difference with variables</li> <li>Introduction to subroutines within written programs</li> </ul>	<ul style="list-style-type: none"> <li>Creating extended pieces of pseudocode for computational problems</li> <li>Creating extended programs from computational problems</li> </ul>	<ul style="list-style-type: none"> <li>Von Neumann created the architecture for the modern computer</li> <li>How the CPU operates</li> <li>The differences between system and application software, and how these benefit the user</li> </ul>	<ul style="list-style-type: none"> <li>Understanding the difference between primary and secondary memory</li> <li>Understand what a network is, and related hardware</li> </ul>
<b>m</b>							

		<ul style="list-style-type: none"> <li>Applying programming constructs into flowcharts and pseudocode</li> </ul>	<ul style="list-style-type: none"> <li>Advantages and disadvantages of different character sets</li> </ul>			<ul style="list-style-type: none"> <li>What system software is such as the operating system and its purposes</li> </ul>	<ul style="list-style-type: none"> <li>Ethical issues within CS that relate to the environment and the economy</li> <li>Laws that relate to CS and their impact</li> </ul>
	<b>Key takeaways for future learning</b>	<ul style="list-style-type: none"> <li>Writing pseudocode using correct conventions</li> <li>Breaking computational problems into inputs, processes and outputs</li> </ul>	<ul style="list-style-type: none"> <li>Convert binary numbers to denary and vice versa</li> <li>Aware of UNICODE and ASCII character sets</li> <li>Shifting binary numbers can divide and multiple</li> </ul>	<ul style="list-style-type: none"> <li>Lists permit users to store multiple pieces of data under one unique identifier</li> <li>How to apply subroutines to computational problems</li> </ul>	<ul style="list-style-type: none"> <li>How to appropriately plan extended computational problems and create these using a written programming language</li> </ul>	<ul style="list-style-type: none"> <li>The CPU has operates on a fetch-execute cycle</li> <li>The operating system has many different purposes</li> </ul>	<ul style="list-style-type: none"> <li>Primary memory is directly accessed by the CPU</li> <li>Secondary memory is not directly accessed by the CPU</li> <li>Advantages and disadvantages of wired vs wireless networks</li> <li>Ethics and laws surrounding CS</li> </ul>
<b>By the end of Key Stage 3, pupils are able to:</b>		<ul style="list-style-type: none"> <li>Understand and apply all computational thinking skills to basic computational problems</li> <li>Understand how computers process inputs and outputs</li> <li>Can use the basic three programming constructs in a block-based and written-based programming language</li> <li>Use different creative software in order to manipulate digital arts</li> <li>Can thinking and describe how modern technology has impacted society, and issues relating to this</li> </ul>					
<b>4 t h  F o r m</b>	<b>Term</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
	<b>Title</b>	Fundamentals of Algorithmic Thinking	Fundamentals of Programming	Advanced Programming	Computer Maths	NEA (Non-exam assessment)	
	<b>Prior Knowledge</b>	<ul style="list-style-type: none"> <li>Writing pseudocode using correct conventions</li> <li>Breaking computational problems into inputs, processes and outputs</li> </ul>	<ul style="list-style-type: none"> <li>How to apply the three programming constructs into a written programming language (Python)</li> <li>Using logic gates to represent boolean logic</li> </ul>	<ul style="list-style-type: none"> <li>How to apply the three programming constructs into a written programming language (Python)</li> <li>Arrays permit users to store multiple pieces of data under one unique identifier</li> </ul>	<ul style="list-style-type: none"> <li>Convert binary numbers to denary and vice versa</li> <li>Aware of UNICODE and ASCII character sets</li> <li>Shifting binary numbers can divide and multiple</li> </ul>	<ul style="list-style-type: none"> <li>Application of all programming constructs to make extended programs individually</li> <li>Data structures are used to store a collection of data</li> <li>Confident use of pseudocode to represent computational problems</li> <li>Being able to read pseudocode and flowcharts for extended pseudocode</li> <li>Files can be open and written on Python</li> <li>A procedure doesn't return a value to a main program but a function does</li> </ul>	
<b>Core Knowledge</b>	<ul style="list-style-type: none"> <li>Using advanced programming constructions within pseudocode and flowcharts</li> </ul>	<ul style="list-style-type: none"> <li>Use of sequence, selection and iteration within programs to complete computational problems</li> <li>Manipulation of strings</li> </ul>	<ul style="list-style-type: none"> <li>Understand how different data structures (1D and 2D arrays and records) can be used in computational problems</li> </ul>	<ul style="list-style-type: none"> <li>Understanding of KB to TB</li> <li>Conversion of binary to hexadecimal and vice versa</li> </ul>	<ul style="list-style-type: none"> <li>Students complete an extended computational project as a piece of coursework for their GCSE without teacher assistance</li> </ul>		

		<ul style="list-style-type: none"> <li>Use of trace tables to follow the sequence of algorithms</li> <li>Understanding and performing searching and sorting algorithms</li> </ul>	<ul style="list-style-type: none"> <li>Application of all programming constructs to make extended programs</li> </ul>	<ul style="list-style-type: none"> <li>How external files can be used within Python</li> <li>The difference between procedures and functions</li> </ul>	<ul style="list-style-type: none"> <li>Understanding of how computers represent data such as sound in binary</li> <li>Techniques of data compression</li> </ul>		
	<b>Key takeaways for future learning</b>	<ul style="list-style-type: none"> <li>Confident use of pseudocode to represent computational problems</li> <li>Being able to read pseudocode and flowcharts for extended pseudocode</li> </ul>	<ul style="list-style-type: none"> <li>Application of all programming constructs to make extended programs individually</li> </ul>	<ul style="list-style-type: none"> <li>Data structures are used to store a collection of data</li> <li>Files can be open and written on Python</li> <li>A procedure doesn't return a value to a main program but a function does</li> </ul>	<ul style="list-style-type: none"> <li>The differences between lossy and loseless compression, and how to create Huffman trees</li> <li>Use of sampling rates to store sound as binary</li> <li>Hexadecimal represent binary in less values</li> </ul>	<ul style="list-style-type: none"> <li>Confident use of pseudocode to represent computational problems</li> </ul>	
<b>5 t h F o r m</b>	<b>Term</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
	<b>Title</b>	Computer Systems	Networks & Cybersecurity	Ethical, legal and environmental impacts of CS	Revision		
	<b>Prior Knowledge</b>	<ul style="list-style-type: none"> <li>The CPU has operates on a fetch-execute cycle</li> <li>The operating system has many different purposes</li> <li>Primary memory is directly accessed by the CPU</li> </ul>	<ul style="list-style-type: none"> <li>Advantages and disadvantages of wired vs wireless networks</li> </ul>	<ul style="list-style-type: none"> <li>Ethics and laws surrounding CS</li> </ul>			

<p><b>Core Knowledge</b></p>	<ul style="list-style-type: none"> <li>• All steps involved in the fetch-execute cycle to complete an instruction</li> <li>• The advantages and disadvantages of different storage systems</li> <li>• What embedded systems are</li> </ul>	<ul style="list-style-type: none"> <li>• Understanding of common network topologies</li> <li>• Knowing the purposes of common network protocols including TCP/IP model</li> <li>• Network safety and methods of security</li> <li>• Different forms of cybersecurity threats and prevention methods</li> </ul>	<ul style="list-style-type: none"> <li>• Students will learn more in-depth knowledge of several different CS topics for essay questions</li> <li>• Topics include cyber security, mobile technologies, theft, copyright, hacking, wearable technologies and privacy</li> </ul>			
<p><b>Key takeaways for future learning</b></p>	<ul style="list-style-type: none"> <li>• The speed of the CPU completing a fetch-execute cycle is known as the clock speed</li> <li>• How solid-state, magnetic and optical disk drives operate</li> <li>• An embedded system has a dedicated function in contrast to PCs</li> </ul>	<ul style="list-style-type: none"> <li>• Know and describe functions of all TCP/IP layers</li> <li>• Bus and star network topologies have unique network setups</li> <li>• A network protocol is a rule followed by the network</li> </ul>	<ul style="list-style-type: none"> <li>• Understanding of general principles rather than just having detailed knowledge on CS issues</li> <li>• Citizens typically value privacy</li> <li>• Governments and security services argue access to private data can keep citizens safe</li> </ul>			
<p><b>By the end of Key Stage 4, pupils are able to:</b></p>		<ul style="list-style-type: none"> <li>• be given a complex computational problem, and use computational thinking skills in order to create a solution using written-based programming language</li> <li>• use pseudocode and flowcharts in order to represent complex computational problems</li> <li>• understand and confidently explain the fetch-execute cycle of a computer</li> <li>• can confidently explain how computers represent data in different forms</li> <li>• describe and explain different network topologies and protocols</li> </ul>				