

### Golden Threads

How to use computational methods to create algorithms solve problems  
 How to use a high-level programming language to turn algorithms into runnable code  
 What are the fundamental principles behind how computers operate?  
 How to analyse, design, develop, test and evaluate a project

### Enrichment

Supporting KS3 computing club  
 After school NEA support

### Review and Evaluation

Summer 2026

	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
Term 1	<b>2.1.1 Thinking abstractly</b> (a) The nature of abstraction (b) The need for abstraction. (c) The differences between an abstraction and reality	(d) Devise an abstract model for a variety of situations. (a) Identify the inputs and outputs for a given situation	Week 7 - Exam question based assessment		Abstraction Algorithm Composition Computational thinking Decomposition Flowchart Pseudocode	Links to: 1.2.3 1.2.4 1.4.2 2.2 2.3
	<b>2.1.2 Thinking ahead</b> (b) Determine the preconditions for devising a solution to a problem (c) The nature, benefits and drawbacks of caching. (d) The need for reusable program components.					
	<b>2.1.3 Thinking procedurally</b>	(a) Identify the components of a problem. (b) Identify the components of a solution to a problem. (c) Determine the order of the steps needed to solve a problem (d) Identify sub-procedures necessary to solve a problem				
	<b>2.1.4 Thinking logically</b>	(a) Identify the points in a solution where a decision has to be taken. (b) Determine the logical conditions that affect the outcome of a decision. (c) Determine how decisions affect flow through a program				



	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
<b>Term 1 (continued)</b>	<p>1.5.1 Computing related legislation</p> <p>(a) The Data Protection Act 1998</p> <p>(b) The Computer Misuse Act 1990.</p> <p>(c) The Copyright Design and Patents Act 1988.</p> <p>(d) The Regulation of Investigatory Powers Act 2000.</p>		Week 4 - Exam question based assessment	There are no laws on the Internet	Artificial Intelligence Artificial Neural Networks Computer Misuse Act Copyright, Designs and Patents Act Creative Commons Data Protection Act Deep learning	Links to 1.3.1
	<p>1.5.2 Moral and ethical Issues</p> <p>The individual moral, social, ethical and cultural opportunities and risks of digital technology:</p> <ul style="list-style-type: none"> <li>Computers in the workforce.</li> <li>Automated decision making.</li> <li>Artificial intelligence.</li> <li>Environmental effects.</li> <li>Censorship and the Internet.</li> <li>Monitor behaviour.</li> <li>Analyse personal information.</li> <li>Piracy and offensive communications.</li> <li>Layout, colour paradigms and character sets.</li> </ul>			Computers always have a negative impact on the environment  Colours have the same meaning in all countries	Encryption Freedom of Information Act Hacking License agreement Machine Learning Malware Reinforcement learning The Regulation of Investigatory Powers Act	



	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
Term 2	<p><b>1.2.4 Types of Programming Language</b></p> <p>(a) Need for and characteristics of a variety of programming paradigms.</p> <p>(b) Procedural languages.</p> <p>(d) Modes of addressing memory (immediate, direct, indirect and indexed).</p>	<p>(c) Assembly language (including following and writing simple programs with the Little Man Computer instruction set).</p> <p>(e) Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation and polymorphism</p>	Week 7 - Exam question based assessment	<p>Procedural programming is the only approach</p> <p>Using global variables is a good approach</p>	<p>Array</p> <p>Inheritance</p> <p>Assembler</p> <p>Low level language</p> <p>Assembly language</p> <p>Machine code</p> <p>Attributes</p> <p>Method</p> <p>Branch</p> <p>Mnemonics</p> <p>Class</p> <p>Object</p> <p>Constructor</p> <p>Polymorphism</p> <p>Encapsulation</p> <p>High level language</p>	<p>Links to:</p> <p>1.1.1</p> <p>1.2.2</p> <p>1.4.1</p> <p>2.2.1</p>
	<p><b>2.2.1 Programming techniques</b></p>	<p>(a) Programming constructs: sequence, iteration, branching.</p> <p>(b) Recursion, how it can be used and compares to an iterative approach.</p> <p>(c) Global and local variables.</p> <p>(d) Modularity, functions and procedures, parameter passing by value and by reference.</p> <p>(e) Use of an IDE to develop/debug a program.</p> <p>(f) Use of object oriented techniques.</p>	Week 7 - Exam question based assessment	<p>Recursion is a good approach in all situations</p> <p>Using global variables is a good approach</p>	<p>Boolean</p> <p>Global</p> <p>Cast</p> <p>Local</p> <p>Character</p> <p>Loop</p> <p>Conditional Statement</p> <p>Parameter</p> <p>Constant</p> <p>Procedure</p> <p>Declaration</p> <p>Python</p> <p>Function</p> <p>Selection</p> <p>Integer</p> <p>Sequence</p> <p>Iteration</p> <p>String</p> <p>Variable</p>	<p>Links to:</p> <p>1.2.2</p> <p>1.2.3</p> <p>1.2.4</p> <p>1.3.4</p> <p>2.3.1</p>
	<p><b>1.1.1 Structure and function of the processor</b></p> <p>(a) The Arithmetic and Logic Unit; ALU, Control Unit and Registers (Program Counter; PC, Accumulator; ACC, Memory Address Register; MAR, Memory Data Register; MDR, Current Instruction Register; CIR). Buses: data, address and control: how this relates to assembly language programs</p> <p>(b) The Fetch-Decode-Execute Cycle; including its effects on registers.</p> <p>(c) The factors affecting the performance of the CPU: clock speed, number of cores, cache.</p> <p>(d) The use of pipelining in a processor to improve efficiency.</p> <p>(e) Von Neumann, Harvard and contemporary processor architecture</p>	<p>Week 6 - Exam question based assessment</p>	<p>The speed of the computer is linearly related to the amount of cache</p>	<p>Accumulator</p> <p>FDE – Fetch Decode Execute cycle</p> <p>ALU – Arithmetic and Logic Unit</p> <p>Interrupt</p> <p>Bus</p> <p>MAR – Memory Address register</p> <p>Cache</p> <p>MDR – Memory Data Register</p> <p>Clock</p> <p>Core</p> <p>PC – Program Counter</p> <p>CPU – Central Processing Unit</p> <p>Pipelining</p> <p>CU – Control Unit</p> <p>Register</p> <p>CIR – Current Instruction Register</p> <p>Von Neumann architecture</p> <p>Word</p>	<p>Links to 1.1.2 and 1.1.3</p>	



	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
Term 3	2.2.1 Programming techniques	(a) Programming constructs: sequence, iteration, branching. (b) Recursion, how it can be used and compares to an iterative approach. (c) Global and local variables. (d) Modularity, functions and procedures, parameter passing by value and by reference. (e) Use of an IDE to develop/debug a program. (f) Use of object oriented techniques.	Week 6 - Exam question based assessment	Recursion is a good approach in all situations Using global variables is a good approach	Boolean Cast Character Conditional Statement Constant Declaration Function Integer Iteration Global Local Loop Parameter Procedure Python Selection Sequence String Variable	Links to: 1.2.2 1.2.3 1.2.4 1.3.4 2.3.1
	1.1.2 Types of processor (a) The differences between and uses of CISC and RISC processors (b) GPUs and their uses (including those not related to graphics). (c) Multicore and Parallel systems.		Week 5 - Exam question based assessment	GPUs are only used for displaying graphics	Harvard Architecture Co-processor Embedded system CISC – Complex Instruction Set Computer RISC – Reduced Instruction Set Computer	Links to 1.1.1 and 1.1.3
	2.1.5 Thinking concurrently a) Determine the parts of a problem that can be tackled at the same time. (b) Outline the benefits and trade offs that might result from concurrent processing in a particular situation.			It is easy to write software that does several different things at the same time		



	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
Term 4	<p><b>2.3.1 Algorithms</b></p> <p>Standard algorithms (bubble sort, insertion sort, binary search and linear search).</p> <p>Standard algorithms (quick sort, binary search).</p>	<p>Implement bubble sort, insertion sort.</p> <p>Implement binary and linear search.</p>	<p>Week 6 - Exam question based assessment</p>	<p>Bubble sort is a useful sorting algorithm</p>	<p>Binary search</p> <p>Bubble sort</p> <p>Divide and conquer</p> <p>Linear search</p> <p>Insertion sort</p> <p>Quick sort</p>	<p>Links to 1.4.2</p> <p>2.2.1</p>
	<p><b>1.1.3 Input, output and storage</b></p> <p>(a) How different input, output and storage devices can be applied to the solution of different problems.</p> <p>(b) The uses of magnetic, flash and optical storage devices</p> <p>(c) RAM and ROM.</p> <p>(d) Virtual storage.</p>		<p>Week 5 - Exam question based assessment</p>		<p>Analogue</p> <p>ADC – Analogue Digital Convertor</p> <p>DAC – Digital Analogue Convertor</p> <p>LCD – Liquid Crystal Display</p> <p>OCR – Optical Character</p> <p>RAM – Random Access Memory</p> <p>ROM – Read Only Memory</p> <p>Recognition</p> <p>Pixel</p> <p>QR code</p> <p>Register</p> <p>Resolution</p> <p>RFID – Radio Frequency ID</p> <p>Sample rate</p> <p>Transistor</p> <p>Volatile</p>	<p>Links to 1.1.1 and 1.1.2</p>
	<p><b>1.2.1 Systems Software</b></p> <p>(a) The need for, function and purpose of operating systems.</p> <p>(b) Memory Management (paging, segmentation and virtual memory).</p> <p>(c) Interrupts, the role of interrupts and Interrupt Service Routines (ISR), role within the Fetch-Decode-Execute Cycle.</p> <p>(d) Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first and shortest remaining time</p> <p>(e) Distributed, embedded, multi-tasking, multi-user and Real Time operating systems.</p> <p>(f) BIOS.</p> <p>(g) Device drivers.</p> <p>(h) Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within another.</p>		<p>Week 5 - Exam question based assessment</p>		<p>Application software</p> <p>CLI – Command Line Interface</p> <p>Device Driver</p> <p>GUI – Graphical User Interface</p> <p>Interrupt</p> <p>Multitasking</p> <p>Operating System</p> <p>Page</p> <p>Peripheral</p> <p>Process</p> <p>Scheduler</p> <p>Segmentation</p> <p>Swapping</p> <p>System software</p> <p>Virtual machine</p> <p>Virtual memory</p> <p>Windows</p>	<p>Links to 1.1.1</p>



	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
Term 5	<p><b>1.4.3 Boolean Algebra</b></p> <p>(a) Define problems using Boolean logic.</p> <p>(d) Using logic gate diagrams and truth tables.</p> <p>(e) The logic associated with D type flip flops, half and full adders.</p>	<p>(b) Manipulate Boolean expressions, including the use of Karnaugh maps to simplify Boolean expressions.</p> <p>(c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.</p>	Exam question based internal assessments for all Y12 content	Boolean logic and simplifying Boolean expressions has no real world use	<p>AND</p> <p>Boolean algebra</p> <p>Flip-flop</p> <p>Full address</p> <p>Half adder</p> <p>Integrated circuit</p> <p>Logic gate</p> <p>Logic circuit</p> <p>NAND</p> <p>NOR</p> <p>NOT</p> <p>OR</p> <p>Truth table</p> <p>XOR</p>	Links to 1.4.1
	<p><b>1.2.2 Applications Generation</b></p> <p>(a) The nature of applications, justifying suitable applications for a specific purpose.</p> <p>(b) Utilities.</p> <p>(c) Open source vs closed source.</p> <p>(d) Translators: Interpreters, compilers and assemblers.</p> <p>(e) Stages of compilation (lexical analysis, syntax analysis, code generation and optimisation).</p> <p>(f) Linkers and loaders and use of libraries.</p>		Exam question based internal assessments for all Y12 content	Open source always means that the software is free	<p>Bytecode</p> <p>Compiler</p> <p>Dynamic Link Library</p> <p>Interpreter</p> <p>Java Virtual Machine</p> <p>Linker</p> <p>Loader</p> <p>Machine code</p> <p>Parsing</p> <p>Translator</p>	Links to 1.2.4 and 2.2.1



	Topics & Substantive Knowledge	Disciplinary Knowledge	Assessment	Misconceptions	Key Vocabulary	Knowledge Tracking
<b>Term 6</b>	<p><b>1.2.3 Software Development</b></p> <p>(a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development.</p> <p>(b) The relative merits and drawbacks of different methodologies and when they might be used.</p> <p>(c) Writing and following algorithms.</p> <p>(d) Different test strategies, including black and white box testing and alpha and beta testing</p> <p>(e) Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation.</p>		Week 6 - Exam question based assessment		<p>Agile model</p> <p>Alpha testing</p> <p>Beta testing</p> <p>Beta version</p> <p>Black box testing</p> <p>Extreme Programming</p> <p>RAD – Rapid Action Development</p> <p>Spiral model</p> <p>Software development lifecycle</p> <p>System testing</p> <p>Testing</p> <p>Unit testing</p> <p>Waterfall model</p> <p>White box testing</p>	Links to 2.1 and 2.2.1
	<p><b>1.3.2 Databases</b></p> <p>(a) Relational database, flat file, primary key, foreign key, secondary key, entity relationship modelling, normalisation and indexing.</p> <p>(b) Methods of capturing, selecting, managing and exchanging data</p> <p>(e) Referential integrity.</p> <p>(f) Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking and redundancy.</p>	<p>(c) Normalisation to 3NF</p> <p>(d) SQL – Interpret and modify.</p>	Week 5 - Exam question based assessment		<p>Atomicity</p> <p>Attribute</p> <p>Boolean</p> <p>Composite primary key</p> <p>Consistency</p> <p>Database</p> <p>Database management system</p> <p>Deadlock</p> <p>Durability</p> <p>Electronic Data Interchange</p> <p>Entity</p> <p>Field</p> <p>First Normal Form</p> <p>Flat-file database</p> <p>Foreign key</p> <p>INT</p> <p>Isolation</p> <p>Non-key dependency</p> <p>Normalisation</p> <p>Operators</p> <p>Partial dependency</p> <p>Primary key</p> <p>Query</p> <p>Record</p> <p>Record locking</p> <p>Redundancy</p> <p>Referential integrity</p> <p>Relational database</p> <p>Second Normal Form</p> <p>Serialisation</p> <p>SQL</p> <p>Structured data</p> <p>Table</p> <p>Third Normal Form</p> <p>Transaction</p> <p>Unstructured data</p> <p>Varchar</p> <p>Wildcard characters</p>	