

Year 9 KNOWLEDGE ORGANISER

KS3 Computing

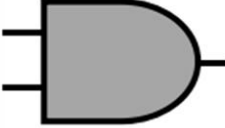
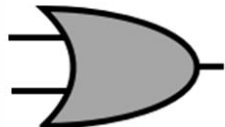
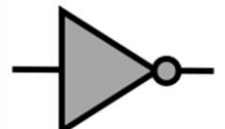


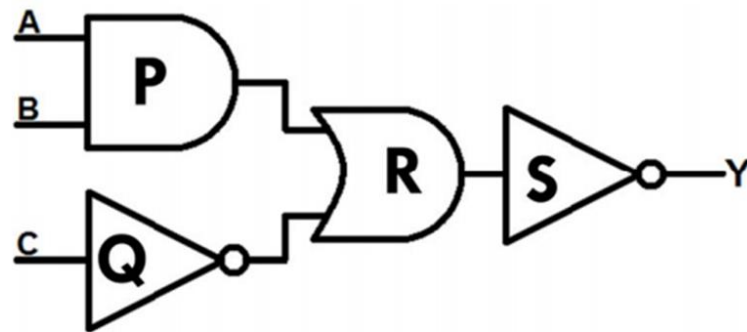
Autumn 1	Computational Thinking - Programming algorithms			
	KNOW	SHOW	1	The Software Development Cycle
	Problem solving needs information and filtering	Pattern recognition in problem solving	2	Problem Decomposition and Algorithm Design
	How to use logic to solve simple problems	How to predict outputs from inputs	3	Testing and success criteria
		How to evaluate effective algorithms	4	Programming Techniques
			5	Program Development/Testing Documentation
			6/7	Program Development (Optional)
Autumn 2	Programming - Further programming techniques in python			
	KNOW	SHOW	1	Effective coding
	How to test solutions	The effect of the scope of a variable	2	Program Development (Optional)
	What a parameter is	Use of a range of operators and expressions in program control	3	Final Testing
	How to read a two dimensional data array	Debug modular programs	4	Evaluation
	How to design a program	How to select iterative loop parameters	5	Evaluation
	How to read iterative loops		6	Performance Review
			7	Assessment

Year 9 KNOWLEDGE ORGANISER

KS3 Computing

Key vocabulary	
Logic Gate	A logic gate is an building block of a digital circuit. Most logic gates have two inputs and one output. At any given moment, every terminal is in one of the two binary conditions 0 or 1.
And	A logic gate which returns a 1 when both inputs are 1's. Else a 0 is returned.
Or	A logic gate which returns 1 when either or both of the inputs are 1.
Not	A logic gate which inverts its input.
Truth Table	A table which shows outputs from a logic gate or circuit given certain inputs.

Binary Logic Gate Diagrams					
AND		Input A			$Q = A \wedge B$
		Input B			
		Output Q			
		0	0	0	
		0	1	0	
OR		Input A			$Q = A \vee B$
		Input B			
		Output Q			
		0	0	0	
		0	1	1	
NOT		Input A		Output Q	$Q = \neg A$
		0		1	
		1		0	



A	B	C	P	Q	R	S/Y
0	0	0	0	1	1	0
0	0	1	0	0	0	1
0	1	0	0	1	1	0
0	1	1	0	0	0	1
1	0	0	0	1	1	0
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	1	0	1	0

Year 9 KNOWLEDGE ORGANISER

KS3 Computing



Spring 1	Computers - Introduction to Linux (Self-Led)			
	KNOW	SHOW	1	Introduction to the Linux Operating System
	How to enter commands into a command line	Input of a range of commands to the command line	2	Command Line
	Some differences between common operating systems	Compare different operating systems	3	File Management
	How to use a range of application software to complete tasks	Use of the command line to model tasks normally done with a GUI.	4	Storing Text files
			5	Editing Files (Optional)
			6	Assessment
Spring 2	IT - Artificial Intelligence (Self-Led)			
	KNOW	SHOW	1	Artificial Intelligence
	What artificial intelligence is	An understanding of some of the ethical issues around artificial intelligence	2	History of Artificial Intelligence
	Give examples		3	AI developments
	Say how it can be useful	Evaluate the impact of technology on society now and in the future	4	Consequences
			5	AI ethics (Optional)
			6	Assessment

Year 9 KNOWLEDGE ORGANISER

KS3 Computing

Computer Science

Key vocabulary	
Antimalware	Software designed to protect a computer in one of 3 ways: preventing installation of harmful software, preventing important files from being changed, scanning for virus activity on the system and removing as appropriate. Antimalware protects against worms, Trojan Horses, spyware, adware and keyloggers.
Antivirus	Software designed to protect against viruses.
Update	New malware is released regularly and so anti-malware definitions must be up-to-date to protect from the latest viruses.
Firewall	Hardware or software designed to prevent unauthorised access to or from a private network or intranet. All messages entering or leaving the network will pass through the firewall to be examined.
Password Protection	In a networked environment such as a school or a company, multiple users use many of the computers. Passwords should be strong (Not easy to guess, lower and uppercase letters, numbers, symbols).
Access Levels	Part of an access control procedure for computer systems, which allows a system administrator to set up a hierarchy of users. Thus, the low-level users can access only a limited set of information.
Encryption	Changing data before transmission so someone can only decipher it with the appropriate key to unlock information. Interceptors would find the message unintelligible.
Key	A cryptographic key is a string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa. This key remains private and ensures secure communication.
Symmetric Key encryption	A secret key algorithm (sometimes called a symmetric algorithm) is a cryptographic algorithm that uses the same key to encrypt and decrypt data.
Asymmetric key encryption	Asymmetric cryptography, also known as public key cryptography, uses public and private keys to encrypt and decrypt data. The keys are simply large numbers that have been paired together but are not identical (asymmetric).

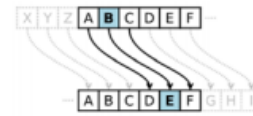


Figure 3 – Symmetric encryption methods such as the Caesar cipher involve shifting letters along the alphabet.

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	F	O	X	A	B	C	D	E	G	H	I	J	K	L	M	N	P	Q	R	S	T	U	V	W	Y	Z

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	B	C	D	E	G	H	I	J	K	L	M	N	P	Q	R	S	T	U	V	W	Y	Z	F	O	X	A

Figure 2 – Keyword encryption involves using a keyword to begin filling up the alphabet, then the rest is filled with remaining letters.

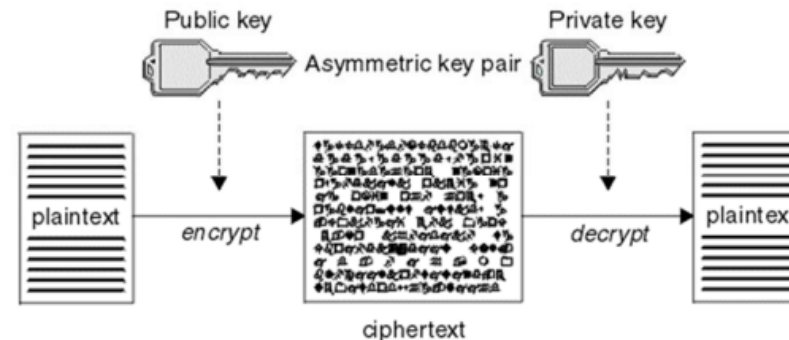


Figure 4 – Asymmetric key encryption uses public keys to encrypt data for somebody who then uses their private key to decrypt it.

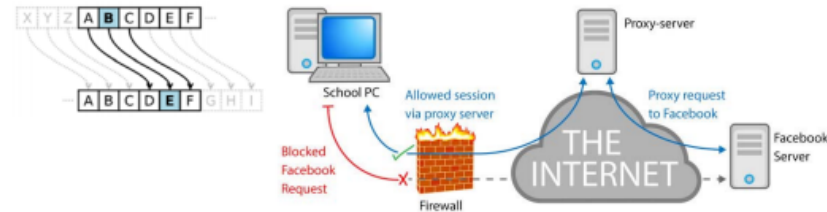


Figure 1 - A firewall sits on the edge of a network and chooses which traffic to allow through using a set of rules. As shown above the rules may not always be strong enough.

Year 9 KNOWLEDGE ORGANISER

KS3 Computing



Summer 1	Networking - Cryptography (Self-Led)			
	KNOW	SHOW	1	Encryption
	What a cipher is and how to decode a simple cipher with a key	An understanding of cyber threats and cyber defence	2	Caesar Cipher
	Why we need to encrypt data	An ability to decode using more complex ciphers and methods such as hashing and salting	3	Hashing
	How information is transmitted over networks		4	Decoding Ciphers
	What a strong password looks like	How to manipulate simple encryption techniques.	5	Encryption ethics
	What public key encryption is		6	Public Key Cryptography (Optional)
			7	Assessment
Summer 2	Data Representation - Boolean Logic and Processing			
	KNOW	SHOW	1	Fetch-Decode-Execute
	How to work out the output of simple logic circuits	How to work out the output of complex logic circuits	2	History of computers
	What processor speed means	The history and projected future of processing speed	3	Code Translators
	What a truth table is and how to interpret it	How bit patterns represent numbers, images and sound.	4	Boolean Logic
			5	Logic Puzzles (Optional)
			6	Emerging and Cloud S/ware
			7	Assessment

Year 9 KNOWLEDGE ORGANISER

KS3 Computing

Key vocabulary	
Operating System	(OS) A collection of programs that tell the hardware what to do. They are necessary on most computer systems, other than simple self-booting systems devoted to a single task.
Kernel	A part of the operating system that connects applications to the hardware. Applications do not have direct control over hardware.
User interface	A system for providing a means of communication between the user and the operating system. Sits above the kernel.
Batch file	A file that contains a series of command line instructions.
DLL	Dynamic Linked Library
Library	A series of instructions and commands available to a programmer.
Firmware	Software that is permanently stored on a device using a ROM device.
Multi-Tasking	If a current program is loading data from a slow peripheral, rather than allow the CPU to stand idle it will turn its attention to another process. Often used to run several programs at one time.
Multi-User OS	An operating system that allows many users to log into the same system simultaneously.
Real time OS	Safety-critical systems that require a short and guaranteed response, such as in a nuclear power plant.
Scheduling	The process of arranging, controlling and optimising commands within the CPU.
User management	A set of controls that allow the access and actions of users to be restricted.
Memory Management	Operating system ensures that all data is stored safely and efficiently in the correct location.
Paging	Splitting programs into equal sized pages to fill available blocks in memory.
Segmentation	Splitting programs into blocks to fill available blocks in memory.
Driver	A program that controls a peripheral device.
Peripheral Management	Managing the communication through signals between a device and its driver.

Roles of an operating system	
Managing hardware & peripherals	
Managing programs installed and being run	
Managing data transfer between memory locations, the CPU and secondary storage	
Providing the interface between the hardware and the applications	
Providing an interface between the computer and user, managing display to the screen	
Managing security and organising data so that it is not overwritten	
Providing a file system for the storage and retrieval of files	
User interfaces	
Command Line	Commands are given to the operating system through using text. Normally used by technicians as it can provide powerful instructions quickly.
Graphical User Interface (GUI)	The use of small icons that represent applications and actions that are performed to reduce the need to learn commands.
Voice Input	Mainly used on mobile devices to provide voice input to a computer system.
Typical operating systems	
Android	Developed by Google to run on mobile devices, based on Linux.
Mac OS	Apple's mobile device operating system used on iPhones, iPads and Apple TV.
Unix	A widely used platform for building alternative operating systems above including MAC OS X and Linux.
Linux	Available in many distributions, Linux is an open source operating system based on Unix.
Mac OS X	Operating system on Apple computers based on Unix.
Windows	Most commonly used operating system developed by Microsoft. Windows is used on laptop and desktop PCs and on a range of devices.

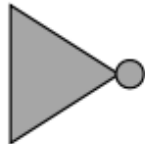
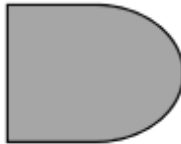
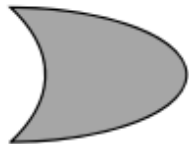
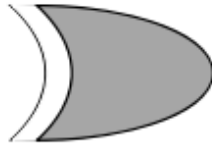
Year 9 KNOWLEDGE ORGANISER

KS3 Computing

Key vocabulary	
Algorithm	A set of instructions which is followed to solve a given problem. Can be represented using a flowchart or Pseudocode.
Abstraction	Removing any unnecessary detail from a problem in order to solve it. Identifies the information that can be removed from the problem without changing it.
Decomposition	Breaking a large problem down with no known solution into smaller steps and stages.
Algorithmic thinking	Algorithmic thinking is a way of getting to a solution through the clear definition of the steps needed – nothing happens by magic.
Searching algorithm	An algorithm for finding values within a set of data.
Linear search	When a list is unsorted and an item needs to be found the algorithm will start at the beginning and move through until it finds the required value.
Binary search	If a list is sorted, an efficient search can be undertaken. It works by repeatedly dividing the set in half and checking where the value is in relation to the current one. It continues until the list has been fully checked or the search term found.
Sorting Algorithm	An algorithm used to sort a set of data into a given order. Examples include bubble sort, insertion sort and merge sort.
Sequencing	Writing steps down in an order in which they must happen.
Selection	Being able to select between different options or scenarios.
Iteration	Iteration is the act of repeating a process, either to generate an unbounded sequence of outcomes, or with the aim of approaching a desired goal, target or result.
Variable	A value, which can change when a program is run. A variable is a memory location. It has a name that is associated with that location; the location stores some data.



Data types		
Integer	A whole number, such as 3, -45, 108	2 or 4 bytes
Real / Float	A number with a fractional part such as 43.69, -9.32.	4 or 8 bytes
Char / Character	A single character where a character can be any letter, digit, punctuation mark or symbol that can be typed.	1 byte
String	Zero or more characters. A string can be null (empty), just one, or several character.	1 byte per character
Boolean	A Boolean variable has the value of True or False.	1 byte

Binary Logic Gate Diagrams																			
NOT			<table><tr><th>A</th><th>Out</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Out	0	1	1	0										
A	Out																		
0	1																		
1	0																		
AND			<table><tr><th>A</th><th>B</th><th>Out</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Out	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	Out																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR			<table><tr><th>A</th><th>B</th><th>Out</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Out	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	Out																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
XOR			<table><tr><th>A</th><th>B</th><th>Out</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Out	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	Out																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	

Year 9 KNOWLEDGE ORGANISER

KS3 Computing

Python -> English	
<code>print('hello!')</code>	Prints a value on screen (in this case, hello!)
<code>input('')</code>	Inputs a value into the computer.
<code>x=input('')</code>	Inputs a value and stores it into the variable x.
<code>x=int(input(''))</code>	Inputs a value into x, whilst also making it into an integer.
<code>print(str(x))</code>	Prints the variable x, but converts it into a string first.
<code>if name == "Fred":</code>	Decides whether the variable 'name' has a value which is equal to 'Fred'.
<code>else:</code>	The other option if the conditions for an if statement are not met (eg. name = 'Bob' when it should be Fred)
<code>elif name == "Tim"</code>	elif (short for else if) is for when the first if condition is not met, but you want to specify another option.
<code>#</code>	# is used to make comments in code – any line which starts with a # will be ignored when the program runs.

Comparative Operators	
<code>==</code>	Equal to
<code>!=</code>	Not equal to
<code>></code>	Greater than
<code><</code>	Less than
<code>>=</code>	Greater than or equal to
<code><=</code>	Less than or equal to

Key vocabulary	
Python	A high level programming language.
Programming	The process of writing computer programs.
Code	The instructions that a program uses.
Sequence	Parts of the code that run in order and the pathway of the program reads and runs very line in order.
Selection	Selects a pathways through the code based on whether a condition is true
Iteration	Code is repeated (looped), either while something is true or for a number of times
Algorithm	A set of rules/instructions to be followed by a computer system
Variable	A value that will change whilst the program is executed. (eg. temperature, speed)
Comparative Operator	When comparing data, an operator is used to solve the equality such as <>, != or ==
Syntax	The punctuation/way that code has to be written so that the computer can understand it. Each programming language has its own syntax.
Data Type	This indicates how the data will be stored. The most common data types are integer, string, and float/real.
String	A collection of letters, numbers or characters. (eg, Hello, WR10 1XA)
Integer	A whole number. (eg. 1, 189)
Float/Real	A decimal number, not a whole number. (eg. 3.14, -26.9)
Boolean	1 of 2 values. (eg. True, False, Yes, No)