Department: *ICT & Computing (Computer Science)*						
Term	Topic/Subject	Assessment Objectives and Knowledge	Skills	Literacy, Numeracy (including wider reading)	Personal Development (SMSC, British Values, Careers, Healthy Living, Citizenship Equality and Diversity, Preparation for next stages	AFL/Summative Assessment
Autumn 1	1.1 Systems Architecture	• Understand what the CPU of a computer does. • Know the components of a CPU. • Know what the registers in a CPU are. • Know the stages of the fetch, execute cycle. • Know what the registers in a CPU are. • Know the stages of the fetch, execute cycle. • Know what factors affect the speed of a CPU. • Know the stages of the fetch, execute cycle. • Know what is meant by the term: 'embedded system'. • Know several examples of embedded systems.	• Begin learning to program in Python. • Understand how to program in Python.	CPU, Von Neumann Architecture, MAR, MDR, PC, Accumulators, ALU, CU, Cache, F-D-E Cycle, Clock Speed, Cache Size, Cores, Embedded System		Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil workbooks. End of topic tests for each topic through exa foundation/craigndave resources & smart revise software.
	1.2 Memory	• Know the difference between RAM and ROM. • Know the purpose of ROM in a computer system. • Know the purpose of RAM in a computer system. • Understand the need for virtual memory. • Understand the purpose of flash memory.		RAM, ROM, Virtual Memory, Flash Memory		
	1.3 Storage	• Understand the need for		Secondary		

		secondary storage. • Know the common types of storage. • Know the characteristics of storage devices. • Understand the suitability of storage devices for given applications. • Understand the advantages and disadvantages of devices based on their characteristics. • Know what data capacity means. • Understand how to calculate data capacity requirements.		storage, Optical storage, Magnetic storage, Solid state storage, Storage capacity, Storage speed, Storage portability, Storage durability, Storage cost		
Autumn 2	1.4 Wired & Wireless Networks	• Know what is meant by 'stand-alone' computers. • Know the different types of networks: LAN and WAN. • Understand the advantages of networking. • Understand the implications of networking. • Know what factors affect the performance of networks. • Know what a client-server model is. • Know what a peer-to-peer model is. • Understand the different roles computers have in each model.		LAN, WAN, Client-server network, Peer-to-peer network, Stand-alone computer, WAP, Router / Switch, NIC, Transmission media, DNS, Hosting, The cloud, Virtual networks		Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil workbooks. End of topic tests for each topic through exa foundation/craigndave resources & smart revise software.

		- Know the hardware needed to connect a LAN. - Understand the purpose of each piece of hardware. - Understand the term DNS (Domain Name Server). - Understand what is meant by the term, 'hosting'. - Understand what is meant by the term, 'cloud'.				
	1.5 Network Topologies	- Know what a star network is. - Know what a mesh network is. - Understand the internet is an example of a partial mesh network. - Know the advantages and disadvantages of star and mesh networks. - Know how WiFi mesh networks are connected. - Know the advantages and disadvantages of wireless mesh networks. - Understand the relationship between WiFi frequencies and channels - Know the basics of how cryptography can work with a simple key. - Know how wireless devices authenticate with each other before communicating data. - Understand the difference		Star network topology, Mesh network topology, WiFi, Frequency, Channels, Encryption, Ethernet, IP Address, MAC address, Protocol, TCP/IP, HTTP, HTTPS, FPT, POP, IMAP, SMTP, Packet switching		

		• between a private key and public keys. • Understand why private (master) keys are never shared. • Know what Ethernet is. • Know what an Ethernet frame is. • Know that Ethernet has become a standard word synonymous with cabled LANs and their hardware. • Understand the implications on Ethernet if twisted pair and switches are used compared to coaxial cable. • Understand the uses of MAC and IP addressing. • Know what 7 popular protocols are used for. • Know the advantages of a packet switching approach on the internet. • Understand how packet switching works. • Understand the different roles of IP and TCP.				
Spring 1	1.6 System Security	• Understand the different forms of attack to computer systems. • Understand the threat from malware. • Understand how to identify and protect against malware. • Understand phishing.		Malware, Phishing, Social engineering, Brute force attacks, Denial of service attacks, Data		Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil

		• Understand how to identify and protect against phishing. • Understand brute force attacks. • Understand how to identify and protect against brute force attacks. • Understand denial of service attacks. • Understand how to identify and protect against denial of service attacks. • Understand data interception and theft as a security threat. • Understand how to identify and protect against data interception. • Understand the concept of SQL injection. • Understand how to protect against SQL injection. • Understand ways in which people are a weak point in secure systems. • Know what is meant by the term, 'network forensics'. • Know what is meant by the term, 'penetration testing'. • Know the implications of a poor network policy.		interception, SQL injection, Network policies, Penetration testing, Network forensics, Anti-malware software, Firewalls, User access level, Password	workbooks. End of topic tests for each topic through exa foundation/craigndave resources & smart revise software.
	1.7 Systems Software	• Know the purpose and functionality of systems		Systems software, User	

		software. • Know the different types of user interface and understand the features of each. • Know what is meant by the term multi-tasking. • Understand how the OS manages the memory. • Understand the need for device drivers. • Understand what is meant by the term, 'user management'. • Understand ways in which the operating system manages files. • Understand encryption utilities. • Understand defragmentation utilities. • Understand data compression utilities. • Understand what is meant by a 'full backup'. • Understand what is meant by an 'incremental' backup.		interface, Memory management, Multitasking, Peripheral management, Device drivers, User management, File management, Utility system software, Encryption software, Defragmentation software, Data compression, Full backup, Incremental backup		
Spring 2	1.8 Ethical, legal, cultural & environmental concerns	• Know a range of things to consider beyond development when implementing new computer systems. • Understand at least one ethical issue of computer technology. • Understand at least one issue related to privacy and computer		Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues, Stakeholder, Open source	Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues, Stakeholder, Open source software, Proprietary software, Data Protection Act, Computer Misuse Act, Copyright Designs and Patents Act, Creative	Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil workbooks.

		technologies • Know the principles of the Acts of Parliament: 　○ Data Protection Act 1998 　○ Computer Misuse Act 1990 　○ Copyright Designs and Parents Act 1988 　○ Freedom of Information Act 2000 • Know what Creative Commons Licensing means. • Understand some of the key cultural issues of computer science: 　○ The impact of technology on our daily lives. 　○ The 'digital divide'. • Globalisation. • Understand the environmental impact of computers in terms of: 　○ Manufacturing 　○ Use • Disposal • Know how to identify key stakeholders. • Know how to consider a scenario from the perspective of the stakeholders. • Understand at least one scenario of the impact of computer science.		software, Proprietary software, Data Protection Act, Computer Misuse Act, Copyright Designs and Patents Act, Creative commons licensing, Freedom of Information Act	commons licensing, Freedom of Information Act	End of topic tests for each topic through exa foundation/craigndave resources & smart revise software.
	2.1 Algorithms	• Know what is meant by the term 'abstraction'.	• Understand how to solve computational	Computational thinking,		

		- Know some examples of abstraction. - Know what is meant by problem decomposition. - Know the advantages of decomposition when applied to programming. - Know an example of problem decomposition.	problems by applying algorithmic thinking. - Understand the linear search algorithm. - Understand it is not an efficient algorithm, but it is easier to program than alternatives and does not require the items to be in any order. - Understand the binary search algorithm. - Know the special condition of the list of items for the binary search to work. - Understand the bubble sort algorithm. - Understand the merge sort algorithm. - Understand the insertion sort algorithm.	Abstraction, Decomposition, Algorithmic thinking, Binary search, Linear search, Bubble sort, Merge sort, Insertion sort, Algorithm, Pseudocode, Flow diagram		
Summer 1	2.2 Programming Techniques	- Know what is meant by the following key terms: - Variables - Constants - Operators - Input - Output - Assignment - Know the 3 programming constructs. - Know the different variable data	- Understand basic string manipulation commands. - Understand how to use basic file handling operations: - Open files - Read from files - Write to files - Close files - Understand how an	Variable, Constant, Inputs, Outputs, Assignments, Sequence, Selection, Iteration, Count controlled loop, Condition controlled loop, String		Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil workbooks. End of topic tests for each topic through exa

| | | types. | array or list can be used to store data.
• Understand that arrays can be one or two dimensional.
• Understand that programs can be structured using procedures and functions.
• Understand the need for casting.
• Know the arithmetic operators.
• Know the Boolean operators.
• Understand that the ALU in the processor handled arithmetic and logic.
• Understand the term 'record'.
• Understand the SQL commands:
  ○ SELECT
  ○ FROM
  ○ WHERE (including the Boolean operators)
  ○ LIKE
• Know the purpose of nested SELECTs. | manipulation, File Handling, SQL, Array/Lists, Subroutine, Procedure, Function, Data type, Integer, Real, Boolean, Character, String, Casting, Arithmetic operator, Boolean operator | | foundation/craigndave resources & smart revise software. |
| | 2.3 Producing Robust Programs | • Know what is meant by the term | • Understand why input sanitisation is | | | |

		"defensive design considerations" when writing programs
		• Know a range of validation techniques that can be used to write a robust program.
		• Know a range of potential problems that can occur when a program is running, especially if it requires communication to servers, peripherals, data in files and arithmetic. Know four reasons why a program should be tested.
		• Know what iterative testing is.
		• Know what final/terminal testing is.
		• Know what a syntax error is.
		• Know what a logic error is.
		• Understand how robust programs are made.
			necessary.
			• Understand why input validation is necessary.
			• Understand some authentication techniques a programmer may choose to use to protect their program from misuse.
			• Understand that because a program works, it doesn't mean it works for all inputs.
			• Understand that suitable test data for a program needs to include:
			  ○ Valid data
			  ○ Invalid data
			  ○ Edge/borderline data
			• Extreme, but valid data
				Defensive design, Input sanitisation / validation, Authentication, Maintainability, Comments, Indentation, Testing, Iterative testing, Final/terminal testing, Syntax errors, Logic errors, Test data		

Summer 2	2.4 Computational Logic	• Know why data is represented in computer systems in binary form.
		• Understand some of the ways this is achieved for different situations:
		  ○ how RAM works
		  ○ how hard drives work
		  ○ how optical drives work
		• Know how to make simple logic diagrams from Boolean expressions using AND, OR, NOT
			Understand how to complete truth tables from one and two level logic diagrams.	Data representation, Binary data representation, logic diagram, AND, OR NOT, Truth table, Computing-related mathematics		Assessment & Feedback through topic based questions, and through practical programming activities. Feedback provided on pupil workbooks.
						
						End of topic tests for each topic through exa foundation/craigndave

	.		• Know how the basic mathematical operators can be used:
			  ○ +
			  ○ -
			  ○ /
			  ○ *
			  ○ ^ Exponentiation (** in Python)
			  ○ DIV Integer division (// in Python)
			  ○ MOD Modulus(% in Python)
							resources & smart revise software.

	2.5 Translators & Facilities of languages		• Know the characteristics of high level and low level programming languages.
			• Understand the terms:
			  ○ Source code
			  ○ Assembly code
			  ○ Machine code
			  ○ Opcode
			  ○ Operand
			• Understand how to write programs in a low level language using assembly with Little Man Computer
			• Know a range of facilities provided by an integrated development environment (IDE) to assist the programmer in writing code.
				Understand the differences between assemblers, compilers and interpreters.	Low level language, High level language, Translators, Assembler, Compiler, Interpreter, IDE, Error diagnostics, Run-time environments	