

Campsmount Academy

‘Creating the best version of you today, for success tomorrow’.

Computer Science	Year 10	HT1	HT2	HT3	HT4	HT5	HT6
	Topic/ Big idea	1.1 - CPU 1.2 - Memory and Storage Practical Programming	1.2 - Data Units 1.3 - Networks Practical Programming	1.3 - Networks 1.4 - Network Security Practical Programming	1.4 - Network Security 1.5 - Systems Software Practical Programming	1.6 - Impacts of Technology Practical Programming	Unit 1 Exam Prep and Revision Responding to Programming Scenarios
	Curriculum Related Expectation	- What actions occur at each stage of the fetch-execute cycle - The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle - The purpose of each register, what it stores (data or address) - The difference between storing data and an address - Understanding of each characteristic listed - The effects of changing any of the common characteristics on system performance, either individually or in combination - What embedded systems are	- Why data must be stored in binary format - Familiarity with data units and moving between each - Data storage devices have different fixed capacities - Calculate required storage capacity for a given set of files - Calculate file sizes of sound, images and text files - sound file size = sample rate x	- A Domain Name Service (DNS) is made up of multiple Domain Name Servers - A DNS's role in the conversion of a URL to an IP address - Concept of servers providing services (e.g. Web server → Web pages, File server → file storage/retrieval) - Concept of clients requesting/using services from a server - The Cloud: remote service provision (e.g. storage, software, processing) - Advantages and disadvantages of the Cloud - Advantages and disadvantages of the	- Understanding of how to limit the threats posed in threats - Understanding of methods to remove vulnerabilities - Knowledge/principles of each prevention method: - What each prevention method may limit/prevent - How it limits the attack - What each function of an operating system does - Features of a user interface - Memory management:	- Technology introduces ethical, legal, cultural, environmental and privacy issues - Knowledge of a variety of examples of digital technology and how this impacts on society - An ability to discuss the impact of technology based around the issues listed - The purpose of each piece	Recap and Revision of Unit 1 Theory (possibly start Unit 2)

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

		- Typical characteristics of embedded systems - Familiarity with a range of different embedded systems - Why computers have primary storage (memory) - How this usually consists of RAM and ROM - Key characteristics of RAM and ROM - Why virtual memory may be needed in a system - How virtual memory works - Transfer of data between RAM and secondary storage when RAM is full - Why computers have secondary storage - Recognise a range of secondary storage devices/media - Differences between each type of storage device/medium - Compare advantages/disadvantages for each storage device	- duration (s) x bit depth - image file size = colour depth x image height (px) x image width (px) - text file size = bits per character x number of characters - enary number range 0 – 255 - Hexadecimal range 00 – FF - Binary number range 00000000 – 11111111 - Understanding of the terms 'most significant bit', and 'least significant bit' - Conversion of any number in these ranges to	- Star and Mesh topologies - Apply understanding of networks to a given scenario - Compare benefits and drawbacks of wired versus wireless connection - Recommend one or more connections for a given scenario - The principle of encryption to secure data across network connections - IP addressing and the format of an IP address (IPv4 and IPv6) - A MAC address is assigned to devices; its use within a network ; its format - The principle of a standard to provide rules for areas of computing - Standards allows hardware/software to interact across different manufacturers/producers - The principle of a (communication)	- The transfer of data between memory - The allocation of memory to applications - How it allows for multitasking - Understand that: - Data is transferred between devices and the processor - This process needs to be managed - User management functions, e.g.: - Allocation of an account - Access rights - Security, etc. - File management, and the key features, e.g.: - Naming - Allocating to folders - Moving files - Understand that computers often come with utility software, and how	- of legislation and the specific actions it allows or prohibits - The need to license software and the purpose of a software licence - Features of open source (providing access to the source code and the ability to change the software) - Features of proprietary (no access to the source code, purchased commonly as off-the-shelf) - Recommend a type of licence for a given scenario including	
--	--	---	--	--	---	---	--

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

		○ Be able to apply their knowledge in context within scenarios ▪ Practical Programming in Python: ○ Data Types ○ Using and Assigning Variables ○ Inputs and Outputs ○ Calculations with Arithmetic and Comparison Operators ○ Selection (If, Elif, Else) ○ Iteration (For)	▪ another number base ▪ Ability to deal with binary numbers containing between 1 and 8 bits ○ e.g. 11010 is the same as 00011010 ▪ Understand the effect of a binary shift (both left or right) on a number ▪ Carry out a binary shift (both left and right) ▪ How characters are represented in binary ▪ How the number of characters stored is limited by the bits available ▪ The differences between and impact of each character set	- protocol as a set of rules for transferring data ▪ That different types of protocols are used for different purposes ▪ The basic principles of each protocol i.e. its purpose and key features ▪ How layers are used in protocols, and the benefits of using layers; for a teaching example, please refer to the 4-layer TCP/IP model ▪ Threats posed to devices/systems ▪ Knowledge/principles of each form of attack including: ○ How the attack is used ○ The purpose of the attack ▪ Practical Programming in Python: ○ Random Module	- this performs housekeeping tasks ▪ Purpose of encryption, defragmentation and data compression software and why it is required ▪ Utility software is needed to perform additional tasks that may not be carried out by an operating system	benefits and drawbacks	
--	--	---	--	--	--	------------------------	--

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

			▪ Understand how character sets are logically ordered, e.g. the character code for 'B' will be one more than the character code for 'A' ▪ Binary representation of ASCII in the exam will use 8 bits ▪ Each pixel has a specific colour, represented by a specific code ▪ The effect on image size and quality when changing colour depth and resolution ▪ Metadata stores additional image information (e.g. height, width, etc.) ▪ Analogue sounds must be stored in binary				
--	--	--	---	--	--	--	--



Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

		▪ Sample rate – measured in Hertz (Hz) ▪ Duration – how many seconds of audio the sound file contains ▪ Bit depth – number of bits available to store each sample (e.g. 16-bit) ▪ Common scenarios where compression may be needed ▪ Advantages and disadvantages of each type of compression ▪ Effects on the file for each type of compression ▪ The characteristics of LANs and WANs including common examples of each ▪ Understanding of different				
--	--	--	--	--	--	--



Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

			factors that can affect the performance of a network, e.g.: ○ Number of devices connected ○ Bandwidth ▪ The tasks performed by each piece of hardware ▪ The concept of the Internet as a network of computer networks ▪ Practical Programming in Python: ○ Iteration (While) ○ Selection (Case and Switch) ○ Advanced Strings and String Manipulation ○ Accessing and Modifying Files				
--	--	--	---	--	--	--	--

Academy

'Creating the best version of you today, for success tomorrow'.

[illegible]

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

						Questions (Challenge Zone)	Carousel, Practice Questions (Challenge Zone)
	Wider Links	Maths: - Binary and Hexadecimal when considering number systems, place value, conversions and addition. - Representing data in computers and working out calculations. Science: - Electronic Circuits and Cabling when considering how hardware connects together and working within a computer device.	Science: - Types of cables and their speed when discussing wired connections. - Electromagnetic waves and data transmission when discussing wireless connections.	History: - Development of the Internet over time and the impact this has had on the world. Business: - Cyber security attacks and the impact this has on companies, including methods of prevention.	Design and Technology: - Human aided design when considering interfaces and how they have made technology easily accessible for use.	Science: - Fossil Fuels and Renewable Energy when considering the impacts technology has on the environment. English: - Creating a written response to a set scenario that discusses a balanced approach to how technology has had an impact. Geography: - Sustainability and environmental impacts. - The Digital Divide and why this is a	



Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

						factor when looking at geographical locations and cultures. - Laws and censorship linked to computers across the world. - Ethics linked to sending e-waste to LEDCs. Business: - The laws and policies linked to computing such as DPA, GDPR, CDPA, CC licences and CMA.	
--	--	--	--	--	--	--	--

Campsmount Academy

‘Creating the best version of you today, for success tomorrow’.

Computer Science	Year 11	HT1	HT2	HT3	HT4	HT5	HT6
	Topic/ Big idea	2.1 Algorithms 2.2 Programming Fundamentals 1.1, 1.2, 1.3 Revision Programming Scenarios	2.2 Programming Fundamentals 2.3 Producing Robust Programs 1.4 and 1.5 Revision Programming Scenarios	2.3 Producing Robust Programs 2.4 Boolean Logic 1.6 Revision Programming Scenarios	2.5 Programming Languages and Integrated Development Environments Programming Scenarios	Unit 1 and Unit 2 Revision	
	Curriculum Related Expectation	▪ Understanding of these principles and how they are used to define and refine problems ▪ Produce simple diagrams to show: ○ The structure of a problem ○ Subsections and their links to other subsections ▪ Complete, write or refine an algorithm using the techniques listed	▪ Practical use of the data types in a high-level language within the classroom ▪ Ability to choose suitable data types for data in a given scenario ▪ Understand that data types may be temporarily changed through casting, and where this may be useful ▪ Practical use of the additional programming techniques in a high-level language within the classroom ▪ Ability to manipulate strings, including: ○ Concatenation ○ Slicing	▪ The difference between testing modules of a program during development and testing the program at the end of production ▪ Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated ▪ Normal test data as data which should be accepted by a program	▪ The differences between high-level and low-level programming languages ▪ The need for translators ▪ The differences, benefits and drawbacks of using a compiler or an interpreter ▪ Knowledge of the tools that an IDE provides ▪ How each of the tools and facilities listed can be used to help a	Revision and Practice of Unit 1 Theory Recap and Revision of Unit 2 Theory	

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

		- Identify syntax/logic errors in code and suggest fixes - Create and use trace tables to follow an algorithm - Use of nesting for selection and iteration - Flow Chart Symbols - Understand the main steps of the algorithm and the segments of code in it - Understand any pre-requisites of an algorithm - Apply the algorithm to a data set - Identify an algorithm if given the code, pseudocode or Exam Reference Language for it - Practical use of the techniques in a high-level	- Arrays as fixed length or static structures - Use of 2D arrays to emulate database tables of a collection of fields, and records - The use of functions - The use of procedures - Where to use functions and procedures effectively - The use of the following within functions and procedures: - local variables/constants - global variables/constants - arrays (passing and returning) - SQL commands: - SELECT - FROM - WHERE - Be able to create and use random numbers in a program - Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values	- without causing errors - Boundary test data as data of the correct type which is on the very edge of being valid - Invalid test data as data of the correct data type which should be rejected by a computer system - Erroneous test data as data of the incorrect data type which should be rejected by a computer system - Ability to identify suitable test data for a given scenario - Ability to create/complete a test plan - Knowledge of the truth tables for each logic gate	- programmer develop a program - Practical experience of using a range of these tools within at least one IDE		
--	--	---	--	---	--	--	--

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

		language within the classroom - Understanding of each technique - Recognise and use the following operators: - Comparison Operators - Arithmetic Operators	- Understanding of how to deal with invalid data in a program - Authentication to confirm the identity of a user - Practical experience of designing input validation and simple authentication (e.g. username and password) - Understand why commenting is useful and apply this appropriately	- Recognition of each gate symbol - Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios - Ability to work with more than one gate in a logic diagram			
	Keywords	Abstraction, Decomposition, Algorithmic, Structure, Diagram, Error, Logic, Flow, Decision, Pseudo, Trace, Search, Binary, Order, Linear, Sort, Split, Merge, Alphabetical, Numerical, Place, Insert, Sequence	Data, Casting, String, Manipulation, Field, Record, SQL, One-Dimensional, Two-Dimensional, Procedure, Function, Parameter, Module, Random, Import	Defensive, Validation, Misuse, Maintain, Structure, Comment, Error, Test, Iterative, Boundary, Invalid, Erroneous, Trace. Values, Variables, Robust Programming, Refine, Boolean, Gate, Circuit, Combine, Complex, Statement, Power Of, Combination, Scenario	Translate, Compile, Interpret, Integrated, Development, Environment		
	Application of knowledge (Assessment)	End of Topic Assessment for 2.1	End of Topic Assessment for 2.2 Unit 1 PPE	End of Topic Assessment for 2.3 and 2.4	End of Topic Assessment for 2.5 Unit 2 PPE	Practice Exam Questions	

Campsmount Academy

'Creating the best version of you today, for success tomorrow'.

	Revision Strategy	Mind Maps and Look, Cover, Write, Check, Carousel, Practice Questions (Challenge Zone)	Mind Maps and Look, Cover, Write, Check, Carousel, Practice Questions (Challenge Zone)	Mind Maps and Look, Cover, Write, Check, Carousel, Practice Questions (Challenge Zone)	Mind Maps and Look, Cover, Write, Check, Carousel, Practice Questions (Challenge Zone)	Mind Maps and Look, Cover, Write, Check, Carousel, Practice Questions (Challenge Zone)	
	Wider Links	Maths: - Logical reasoning and problem solving when using algorithms, including number sequencing. - Searching and sorting algorithms using efficiency and pattern recognition. English: - Written communication when expressing algorithms in different ways.	Maths: - Problem solving when creating programs. - Mathematical operators and symbols that are using in programming scenarios.	Maths: - Truth tables and Boolean logic to work out logic gates and circuits. Design and Technology: - Iterative and Final testing on a product before launch.	Solely linked to programming and how this is conducted in a professional career.		