Key Stage 4 Framework for Learning

Year 10 2018-2019: Successful Foundations

Curriculum Area: Computing and Technology

Year 10	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Syllabus	2.4 Computational Logic 2.6 Data Representation	1.8 Ethical, legal, cultural and environmental concerns	1.1 Systems Architecture 1.2 Memory 1.3 Storage 1.7 Systems Software	1.4 Wired and Wireless Networks 1.5 Network topologies, protocols and layers 1.6 System Security	2.1 Algorithms	2.2 Programming Techniques 2.3 Producing Robust Programs 2.4 Computational Logic 2.5 Translators and Facilities of Languages 2.6 Data Representation
Knowledge	2.4 – Computational Logic; why data is represented in computer systems in binary form, simple logic diagrams using the operations AND, OR and NOT, truth tables, combining Boolean operators using AND, OR and NOT to two levels, applying logical operators in appropriate truth tables to solve problems, applying computing-related mathematics (+, -, /, *, exponentiation (^), MOD, DIV. 2.6 – Data Representation; units (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte) and how data needs to be converted into a binary format to be processed by a computer, numbers (how to convert positive denary to 8 bit binary and vice versa), adding two 8 bit binary integers and explain overflow errors, binary shifts, converting positive denary numbers to 2 digit hexadecimal numbers, convert from binary to hexadecimal equivalents,	1.8 – Ethical, legal, cultural and environmental concerns; how to investigate and discuss CS technologies while considering (ethical, legal, cultural, environmental and privacy issues), hoe key stakeholders are affected by technologies, environmental impact of CS, cultural implications of CS, open source vs proprietary software, legislation relevant to CS (DPA 1998, CMA 1990, CDPA 1988, Creative Commons, FIA 2000).	1.1 – Systems Architecture; the purpose of the CPU, on Neumann Architecture (MAR, MDR, PC, AC), common CPU components (ALU, CU, Cache), function of the CPU as fetch and execute instructions, common characteristics of CPUs (speed, size, cores), embedded systems. 1.2 – Memory; difference between RAM and ROM, purpose of RAM, purpose of ROM, virtual memory, flash memory. 1.3 – Storage; the need for secondary storage, data capacity and calculation, common types of storage (optical, magnetic, solid state), suitable storage devices and advantages/disadvantages (capacity, speed, portability, durability, reliability, cost). 1.7 – Systems Software; purpose and functionality of systems software, operating systems (user interface, memory, multitasking,	1.4 – Wired and Wireless Networks; types of networks (LAN, WAN), factors that affect the performance of networks, different roles of computers in client-server and peer-to-peer, hardware needed to connect into a LAN (WAP, routers/switches, NIC, transmission media), internet as a worldwide collection of networks (DNS, hosting, cloud), virtual networks. 1.5 – Network topologies, protocols and layers; star and mesh topologies, Wi-Fi (frequency, channels, encryption), Ethernet, uses of IP, MAC and protocols (TCP/IP, HTTP, HTTPAS, FTP, POP, IMAP, SMTP), concept of layers, packet switching. 1.6 – System Security; forms of attack, threats posted to networks (malware, phishing, people as weak points, brute force attacks, data interception and theft, concept of SQL injection, poor network policy), identifying and preventing vulnerabilities	2.1 – Algorithms; computational thinking (abstraction, decomposition, algorithmic thinking), standard searching algorithms (binary and linear search), standard sorting algorithms (bubble, merge and insertion sort), producing algorithms (pseudocode, flow diagrams), interpret correct or complete algorithms.	2.2 – Programming Techniques; uses of variables, constants, operators, inputs, outputs and assignments. Three basic programming constructs (sequence, selection, iteration), basic string manipulation, basic file handling operations (open, read, write, close), use of records to store data, use of SQL to search for data, the use of arrays when solving problems, including bothone and two dimensional arrays. How to use sub programs (functions and procedures) to produce structured code, data types (integer, real, Boolean, character and string, casting), common arithmetic operators, common Boolean operators. 2.3 – Producing Robust Programs; defensive design considerations (input sanitization/validation, planning for contingencies, anticipating misuse, authentication), maintainability (comments, indentation), the purpose of testing, types of testing (iterative, final/terminal), how to identify syntax and

	check digits. Characters (the use of binary codes to represent characters, the term character-set, relationship between the number of bits per character and how characters can be represented). Images (how an image is represented as pixels, metadata, effect of colour depth), sound (sampling and storing, intervals such as size, bit rate and frequency), compression (the need and types (lossy, lossless)).			peripheral management, drivers, user management, file management), utility system software (encryption software, defragmentation, data compression, role and methods of backup (full, incremental).		(penetration testing, network forensics an d policies, anti-malware, firewalls, user access levels, passwords, encryption).		logic errors, selecting and using suitable test data. 2.4 – Computational Logic; why data is represented in computer systems in binary form, simple logic diagrams using the operations AND, OR and NOT, truth tables, combining Boolean operators using AND, OR and NOT to two levels, applying logical operators in appropriate truth tables to solve problems, applying computing-related mathematics (+, -, /, *, exponentiation (^), MOD, DIV. 2.5 – Translators and facilities of languages; characteristics and purpose of different levels of programming language, including low level languages, purpose of translators, characteristics of an assembler, compiler and interpreter. Common tools and facilities available in an IDE (editors, error diagnostics, run-time environment, translators). 2.6 – Data Representation; units (bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte) and how data needs to be converted into a binary format to be processed by a computer, numbers (how to convert positive denary to 8 bit binary and vice versa), adding two 8 bit binary integers and explain overflow errors, binary shifts, converting positive denary numbers to 2 digit hexadecimal numbers, convert from binary to hexadecimal equivalents, check digits. Characters (the use of binary codes to represent characters, the term character-set,

							relationship between the number of bits per character and how characters can be represented). Images (how an image is represented as pixels, metadata, effect of colour depth), sound (sampling and storing, intervals such as size, bitrate and frequency), compression (the need and types (lossy, lossless)).
Skills	2.4 Computational Logic; to understand why data is represented in binary form, simple logic diagrams using the operations AND, OR and NOT, truth tables, combining Boolean operators using AND, OR and NOT to two levels, applying logical operators in appropriate truth tables to solve problems and to apply computing-related mathematics. 2.6 Data Representation; to understand units, numbers, characters, images, sound, compression.	1.8 Ethical, legal, cultural and environmental concerns; to understand and examine how to investigate and discuss Computer Science technologies, how key stakeholders are affected by technologies, the environmental impact of Computer science, the cultural implications of Computer Science, open source vs proprietary software, legislation relevant to Computer Science.		1.1 Systems Architecture; to discover the CPU and the VNA, with common CPU components and functions which relate to the fetch execute cycle. Investigate the common characteristics of CPUs and embedded systems. 1.2 Memory; to understand the difference between RAM and Rom, their purpose, the need for virtual memory and flash memory. 1.3 Storage; to understand secondary storage, data capacities and calculations with common types of storage and storage devices especially their characteristics. 1.7 Systems Software; to understand the purpose and functionality of systems software, operating systems and utility system software.		1.4 Wired & Wireless Networks; understanding the types of networks and factors that affect the performance of networks including the different roles of computers in a client-server and peer-to-peer network. To examine the hardware needed to connect stand-alone computers into a LAN and the internet as a worldwide collection of computer networks with the concept of virtual networks being explored. 1.5 Network topologies, protocols and layers; to understand star and mesh topologies, Wi-Fi, Ethernet, the uses of IP addressing, MAC addressing and protocols. To examine the concept of layers and packet switching. 1.6 System Security; to learn the forms of attacks, threats posed to networks and to identify and prevent vulnerabilities.	2.1 Algorithms; to examine computational thinking, searching and sorting algorithms, how to produce algorithms and interpret, correct or complete algorithms.	2.2 Programming Techniques; to understand the use of variables, constants, operators, inputs, outputs and assignments. To learn about the use of the three basic programming constructs, string manipulation, file handling operations, use of records to store data, use of SQL to search for data, use of arrays when solving problems including one and two dimensional arrays. TO examine how to use sub programs, data types, arithmetic operators and Boolean operators. 2.3 Producing Robust Programs; to understand defensive design considerations, maintainability, purpose of testing, types of testing, how to identify syntax and logic errors, selecting and using suitable test data. 2.4 Computational Logic; to understand why data is represented in binary form, simple logic diagrams using the operations AND, OR and NOT, truth tables, combining Boolean operators using AND, OR and NOT to two levels, applying logical operators in appropriate truth tables to solve problems and to apply computing-related mathematics.

							2.5 Translators and Facilities of languages; to understand the characteristics and purposes of different levels of programming language, the purpose of translators, characteristics of an assembler and compiler and interpreter, common tools and facilities available in IDEs. 2.6 Data Representation; to understand units, numbers, characters, images, sound, compression.
Assessment	**Marking Point 1** Exam question on logic. *Page 75 Revision Workbook (Q4)* **Marking Point 2** Binary Numbers exam question Page 78 Q4 Revision Workbook Hexadecimal equation (Q3) Page 80 Revision workbook **Marking Point 3** Mini exam paper focusing on 2.4/2.6. *Page 55-73, OCR revision guide.* *Page 44-52, 140-156, OCR student book.* **Peer/Self Assessed** Exam questions based on computational logic (2.4)	**Marking Point 1** Exam questions on ethical, legal, cultural and environmental concerns. *Page 25-31, OCR revision guide.* *Page 217-232, OCR student book.* **Marking Point 2** Scenario given, to suggest best course of action based on different law concerns. *Page 25-31, OCR revision guide.* *Page 217-232, OCR student book.* **Self or Peer-Assessment** Exam questions on open source vs proprietary software, *Page 25-31, OCR revision guide.* *Page 181-186, OCR student book.*	**Marking Point 1** Draw and label the VNA with all components and functions. *Page 3, OCR revision guide.* *Page 162-169, OCR student book.* **Marking Point 2** Case study on different memory and storage components – to compare/conclude which is best. *Page 4-7, OCR revision guide.* *Page 174-176, OCR student book.* **Marking Point 3** Quizizz on systems software, self-assessed automatically. *Page 8-10, OCR revision guide.* *Page 162-180, OCR student book.*	**Marking Point 1** Exam questions based on LAN/WAN networks. *Page 13, OCR revision guide.* *Page 187-204, OCR student book.* **Marking Point 2** Draw the different topologies and correctly label. *Page 16, OCR revision guide.* *Page 194-198, OCR student book.* **Self or Peer-Assessment** Scenario given for students to give best system security resolution. *Page 21, OCR revision guide.* *Page 187-204, OCR student book.*	**Marking Point 1** Exam questions on logic. *Page 64-65, OCR revision guide.* *Page 44, OCR student book.* **Marking Point 2** Scenario given to design logic tables. *Page 64-65, OCR revision guide.* *Page 45-49, OCR student book.* **Marking Point 3** Quizizz on data representation, self-assessed automatically. *Page 55-73, OCR revision guide.* *Page 44-52, 140-156, OCR student book.*	**Marking Point 1** Exam questions based on programming techniques. *Page 41-45, OCR revision guide.* *Page 157-161, OCR student book.* **Marking Point 2** Scenario given, to identify errors, test and select suitable information. *Page 57-63, OCR revision guide.* *Page 1-25, OCR revision guide.* **Self or Peer-Assessment** Scenario given in Python, with "cheat sheets" – students to compile working small program. *Page 57-63, OCR revision guide.* *Page 157-161, OCR student book.*
Cultural Enrichment	**READ** Page 64-73, OCR revision guide. **WATCH** 2.4 Computational Logic 2.6 Data Representation **VISIT** Visit MOSI (Museum of	**READ** Page 25-30, OCR revision guide. **WATCH** 1.8 Ethical, Legal, Cultural and Environmental Concerns **VISIT**	**READ** Page 1-12, OCR revision guide. **WATCH** 1.1 Systems Architecture 1.2 Memory 1.3 Storage 1.7 Systems Software	**READ** Page 13-23, OCR revision guide. **WATCH** 1.4 wired & Wireless Networks 1.5 Network Topologies, Protocols and Layers 1.6 System Security	**READ** Page 64-73, OCR revision guide. **WATCH** 2.4 Computational Logic **VISIT** Visit MOSI (Museum of Science and Industry) to	**READ** Page 33-45, OCR revision guide. **WATCH** 2.1 Algorithms 2.2 Programming Techniques 2.3 Producing Robust Programs 2.5 Translators and Facilities

	Science and Industry) to look at how computing has helped shape science and industry.	UKFAST employee to visit CHS to deliver a session on 1.8 (especially security).	**VISIT** School IT Technicians tour or speak around how they use memory, storage and systems software.	**VISIT** Organise trip to UKFAST for students to see how they work with networks, topologies and system security.	look at how computing has helped shape science and industry.	of Languages 2.6 Data Representation **VISIT** Visit Barclays in Knutsford to look at the different work they do – all relating to the topics students have covered.
Character	 QofS – Optimism Students will plan, prepare and deliver a whole range of tasks on the topics above.	 QofS – Empathy Students will consider the impact 1.8 has on society and the benefits/drawbacks of each.	  QofS – Creativity & Curiosity Students will be planning, preparing and different tasks on the topics above.	  QofS – Responsibility & Reflection Students will be responsible for their learning, in and out of school, to ensure they reflect on areas for improvement.	  QofS – Practice & Resiliency Students will be conducting own work throughout, whilst practicing exam questions.	 QofS – Motivation Students will show motivation in their final test of the year.