Computer Science GCSE Edexcel REVISION **CHECK LIST**

INSTRUCTIONS

Paper 1:

1. Work through each topic, one by one.
2. RAG the topics. If it is green, you will be able to answer exam questions on it.
3. For any sub-topic on amber or red, re-watch the smart revise video/read your Cornell notes/use the knowledge organisers (Knowledge organisers) or your student revision guide (if you bought one). Update your RAG as you do this.
4. Once you whole topic is green, go to Smart Revise. Re-do that smart revise topic (the multiple-choice quizzes) Your target is 80%. Your score WILL DROP before it goes back up. This is normal!
5. Once on 80%, try the topic exam questions tasks on Smart Revise.
6. Exam paper practice time. Better to print it off for paper 1. (You will need to be signed in to your school account): EXAM PRACTICE

Paper 1: Principles of Computer Science (*Paper code: 1CP2/01)

Content overview This paper will assess Topics 1 to 5.

Topic 1: Computational thinking

https://student.craigndave.org/videos/gcse-edexcel-topic-1a-computational-thinking-and-algorithms

https://student.craigndave.org/videos/gcse-edexcel-topic-1b-computational-logic

SET YOUR SMART REVISE FILTER TO TOPICS 1A AND 1B.

Understanding of what algorithms are, what they are used for and how they work; ability to follow, amend and write algorithms; ability to construct truth tables.			☹	😐	☺
1.1 Decomposition and abstraction	1.1.1	understand the benefit of using decomposition and abstraction to model aspects of the real world and analyse, understand and solve problems			
	1.1.2	understand the benefits of using subprograms			
1.2 Algorithms	1.2.1	be able to follow and write algorithms (flowcharts, pseudocode*, program code) that use; sequence selection repetition (count-controlled, condition-controlled) and iteration (over every item in a data structure), and input, processing and output to solve problems			
	1.2.2	understand the need for and be able to follow and write algorithms that use; variables constants one- and two-dimensional data structures (strings, records, arrays)			
	1.2.3	understand the need for and be able to follow and write algorithms that use; arithmetic operators (addition, subtraction, division, multiplication, modulus, integer division, exponentiation) relational operators (equal to, less than, greater than, not equal to, less than or equal to, greater than or equal to) and logical operators (AND, OR, NOT)			

	1.2.4	be able to determine the correct output of an algorithm for a given set of data, and use a trace table to determine what value a variable will hold at a given point in an algorithm			
	1.2.5	understand types of errors that can occur in programs (syntax, logic, runtime) and be able to identify and correct logic errors in algorithms			
	1.2.6	understand how standard algorithms work; Bubble sort. merge sort. linear search binary search			
	1.2.7	be able to use logical reasoning and test data to evaluate an algorithm's fitness for purpose and efficiency (number of compares, number of passes through a loop, use of memory)			
1.3 Truth tables	1.3.1	be able to apply logical operators (AND, OR, NOT) in truth tables with up to three inputs to solve problems			

Topic 2: Data

https://student.craigndave.org/videos/gcse-edexcel-topic-2a-binary

https://student.craigndave.org/videos/gcse-edexcel-topic-2b-data-representation-compression

SET YOUR SMART REVISE FILTER TO TOPICS 2A AND 2B

understanding of binary, data representation, data storage and compression.			☹️	😐	😊
2.1 Binary	2.1.1	understand that computers use binary to represent data numbers, text, sound, graphics) and program instructions and be able to determine the maximum number of states that can be represented by a binary pattern of a given length			
	2.1.2	understand how computers represent and manipulate unsigned integers and two's complement signed integers			
	2.1.3	be able to convert between denary and 8-bit binary numbers (0 to 255, -128 to +127)			
	2.1.4	be able to add together two positive binary patterns and apply logical and arithmetic binary shifts			
	2.1.5	understand the concept of overflow in relation to the number of bits available to store a value			
	2.1.6	understand why hexadecimal notation is used and be able to convert between hexadecimal and binary			
2.2 Data representation	2.2.1	understand how computers encode characters using 7-bit ASCII			
	2.2.2	understand how bitmap images are represented in binary (pixels, resolution, colour depth)			
	2.2.3	understand how analogue sound is represented in binary (amplitude, sample rate, bit depth, sample interval)			

	2.2.4	understand the limitations of binary representation of data when constrained by the number of available bits			
2.3 Data storage and compression	2.3.1	understand that data storage is measured in binary multiples (bit, nibble, byte, kibibyte, mebibyte, gibibyte, tebibyte) and be able to construct expressions to calculate file sizes and data capacity requirements			
	2.3.2	understand the need for data compression and methods of compressing data (lossless, lossy)			

Topic 3: Computers

https://student.craigndave.org/videos/gcse-edexcel-topic-3a-hardware

https://student.craigndave.org/videos/gcse-edexcel-topic-3b-software-programming-languages

SET YOUR SMART REVISE FILTER TO TOPICS 3A AND 3B

understanding of hardware and software components of computer systems and characteristics of programming languages.			😟	😐	😊
3.1 Hardware	3.1.1	understand the von Neumann stored program concept and the role of main memory (RAM), CPU (control unit, arithmetic logic unit, registers), clock, address bus, data bus, control bus in the fetch-decode-execute cycle:			
	3.1.2	understand the role of secondary storage and the ways in which data is stored on devices (magnetic, optical, solid state)			
	3.1.3	understand the concept of an embedded system and what embedded systems are used for			
3.2 Software	3.2.1	understand the purpose and functionality of an operating system (file management, process management, peripheral management, user management)			
	3.2.2	understand the purpose and functionality of utility software (file repair, backup, data compression, disk defragmentation, anti-malware)			
	3.2.3	understand the importance of developing robust software and methods of identifying vulnerabilities (audit trails, code reviews)			
3.3 Programming languages	3.3.1	understand the characteristics and purposes of low-level and high-level programming languages			
	3.3.2	understand how an interpreter differs from a compiler in the way it translates high-level code into machine code			

Topic 4: Networks

https://student.craigndave.org/videos/gcse-edexcel-topic-4-networks-and-network-security

SET YOUR SMART REVISE FILTER TO TOPIC 4

Computer Science GCSE Edexcel REVISION **CHECK LIST**

understanding of computer networks and network security.			😟	😐	😊
4.1 Networks	4.1.1	understand why computers are connected in a network			
	4.1.2	understand different types of networks (LAN, WAN)			
	4.1.3	understand how the internet is structured (IP addressing, routers)			
	4.1.4	understand how the characteristics of wired and wireless connectivity impact on performance (speed, range, latency, bandwidth)			
	4.1.5	understand that network speeds are measured in bits per second (kilobit, megabit, gigabit) and be able to construct expressions involving file size, transmission rate and time			
	4.1.6	understand the role of and need for network protocols (Ethernet, Wi-Fi, TCP/IP, HTTP, HTTPS, FTP) and email protocols (POP3, SMTP, IMAP)			
	4.1.7	understand how the 4-layer (application, transport, internet, link) TCP/IP model handles data transmission over a network			
	4.1.8	understand characteristics of network topologies (bus, star, mesh)			
4.2 Network security	4.2.1	understand the importance of network security, ways of identifying network vulnerabilities (penetration testing, ethical hacking) and methods of protecting networks (access control, physical security, firewalls)			

Topic 5: Issues and impact

https://student.craigndave.org/videos/gcse-edexcel-topic-5-issues-and-impact

SET YOUR SMART REVISE FILTER TO TOPIC 5

awareness of emerging trends in computing technologies, and the impact of computing on individuals, society and the environment, including ethical, legal and ownership issues			😟	😐	😊
5.1 Environmental	5.1.1	understand environmental issues associated with the use of digital devices (energy consumption, manufacture, replacement cycle, disposal)			
5.2 Ethical and legal	5.2.1	understand ethical and legal issues associated with the collection and use of personal data (privacy, ownership, consent, misuse, data protection)			
	5.2.2	understand ethical and legal issues associated with the use of artificial intelligence, machine learning and robotics (accountability, safety, algorithmic bias, legal liability)			
	5.2.3	understand methods of intellectual property protection for computer systems and software (copyright, patents, trademarks, licencing)			
5.3 Cybersecurity	5.3.1	understand the threat to digital systems posed by malware (viruses, worms, Trojans, ransomware, key loggers) and how hackers exploit technical vulnerabilities (unpatched software, out-of-date anti-malware) and use social engineering to carry out cyberattacks			
	5.3.2	understand methods of protecting digital systems and data (anti-malware, encryption, acceptable use policies, backup and recovery procedures)			

Computer Science GCSE Edexcel REVISION **CHECK LIST**

Paper 2: Application of Computational Thinking (*Paper code: 1CP2/02)

INSTRUCTIONS

Paper 2:

1. RAG topic 6. If it is green, you will be able to answer exam questions on it or use it in a program.
2. For any sub-topic on amber or red, re-watch the smart revise video/read your Cornell notes or read your student revision guide (if you bought one). Update your RAG as you do this.
3. Once you whole topic is green, go to Smart Revise. Re-do that smart revise topic (the multiple-choice quizzes) Your target is 80%. Your score WILL DROP before it goes back up. This is normal!
4. Once on 80%, try the topic exam questions tasks on Smart Revise.
5. Exam paper practice time. For Paper 2, make a copy of the folder before starting. (You will need to be signed in to your school account): EXAM PRACTICE

https://student.craigndave.org/videos/gcse-edexcel-topic-6a-developing-code

https://student.craigndave.org/videos/gcse-edexcel-topic-6b-programming-fundamentals

SET YOUR SMART REVISE FILTER TO TOPICS 6A AND 6B

This paper will assess Topic 6: Problem solving with programming. The main focus of this paper is:

• understanding what algorithms are, what they are used for and how they work in relation to creating programs • understanding how to decompose and analyse problems • ability to read, write, refine and evaluate programs.			😟	😐	😊
6.1 Develop code	6.1.1	be able to use decomposition and abstraction to analyse, understand and solve problems			
	6.1.2	be able to read, write, analyse and refine programs written in a high-level programming language			
	6.1.3	be able to convert algorithms (flowcharts, pseudocode*) into programs			
	6.1.4	be able to use techniques (layout, indentation, comments, meaningful identifiers, white space) to make programs easier to read, understand and maintain			
	6.1.5	be able to identify, locate and correct program errors. (logic, syntax, runtime)			
	6.1.6	be able to use logical reasoning and test data to evaluate a program's fitness for purpose and efficiency (number of compares, number of passes through a loop, use of memory)			