Introduction

This mapping document highlights the overlapping content between the Edexcel International GCSE Computer Science and the CIE International GCSE Computer Science 0478 for exams in 2026, 2027 and 2028.

Page 1 is this Introduction
Page 2 - 6 presents the full content for the Edexcel iG Computer Science
Page 7 - 10 presents the full content for the CIE International GCSE

Rows highlighted in pale yellow have not mapped.

Key differences	Edexcel	CIE
Languages supported	Python 3, Java, C Family	Python, Java, VB.Net (advised)
Paper 1	2hr Exam	1hr 45min Exam
Paper 2	3hr hybrid (digital/paper)	1hr 45min Exam
Paper 2	3 day window	
Databases	Not in	Present
Logic Gates	Not in	Present
Sensors	Not in	Present
Robotics	Not in	Present
Digital Currency/Blockchain	Not in	Present
Methods of error detection	Not in	Present
Program development lifecycle	Not in	Present
Environment, Ethical Legal issues	Present	Not in
Bus, ring , star mesh topologies	Present	Not in
Network types	Present	Not in
Programming detail	Present	Less content

Subject Content	iG	CIE
Spec points	88	96
Exacty the same or similar	57%	68%

Topic	Sub Topic	International GCSE Reference	International GCSE Learning Objective
Problem Solving	Algorithms	1.1.1	Understand what an algorithm is, what algorithms are used for and be able to interpret algorithms (flowcharts, pseudocode, written descriptions, program code).
		1.1.2	Understand how to create an algorithm to solve a particular problem, making use of programming constructs (sequence, selection, iteration) and using appropriate conventions (flowchart, pseudocode, written description, draft program code).
		1.1.3	Understand the purpose of a given algorithm and how an algorithm works.
		1.1.4	Understand how to determine the correct output of an algorithm for a given set of data.
		1.1.5	Understand how to identify and correct errors in algorithms, including using trace tables.
		1.1.6	Understand how to code an algorithm in a high-level language.
		1.1.7	Understand how the choice of algorithm is influenced by the data structures and data values that need to be manipulated.
		1.1.8	Understand how standard algorithms work (bubble sort, merge sort, linear search, binary search).
		1.1.9	Be able to evaluate the fitness for purpose of algorithms in meeting specified requirements efficiently, using logical reasoning and test data.
	Decomposition and abstraction	1.2.1	Be able to analyse a problem, investigate requirements (inputs, outputs, processing, initialisation) and design solutions.
		1.2.2	Be able to decompose a problem into smaller sub-problems.
		1.2.3	Understand how abstraction can be used effectively to model aspects of the real world.
		1.2.4	Be able to program abstractions of real-world examples.
ing	Develop code	2.1.1	Be able to program abstractions of real-world examples.
		2.1.2	Understand the benefit of producing programs that are easy to read and be able to use techniques (comments, descriptive names (variables, constants, subprograms), indentation) to improve readability and to explain how the code works.
		2.1.3	Be able to differentiate between types of error in programs (logic, syntax, runtime).
		2.1.4	Be able to design and use test plans and test data (normal, boundary, erroneous).
		2.1.5	Be able to interpret error messages and identify, locate and fix errors in a program.
		2.1.6	Be able to determine what value a variable will hold at a given point in a program (trace table).
		2.1.7	Be able to determine the strengths and weaknesses of a program and suggest improvements.
	Constructs	2.2.1	Understand the structural components of a program (variable and type declarations, command sequences, selection, iteration, data structures, subprograms).
		2.2.2	Be able to use sequencing, selection and iteration constructs in their programs.
		2.3.1	Understand the need for, and understand how to use, data types (integer, real, Boolean, char, string).

Topic	Sub Topic	International GCSE Reference	International GCSE Learning Objective
Programming	Data types and structures	2.3.2	Understand the need for, and understand how to use, data structures (records, one-dimensional arrays, two-dimensional arrays).
		2.3.3	Understand the need for, and how to manipulate, strings.
		2.3.4	Understand the need for, and how to use, variables and constants.
		2.3.5	Understand the need for, and how to use, global and local variables when implementing subprograms.
	Input / Output	2.4.1	Understand how to write code that accepts and responds appropriately to user input.
		2.4.2	Understand the need for, and how to implement, validation
		2.4.3	Be able to write code that reads/writes from/to a text file.
	Operators	2.5.1	Understand the purpose of, and how to use, arithmetic operators (add, subtract, divide, multiply, modulus, integer division).
		2.5.2	Understand the purpose of, and how to use, relational operators (equal to, less than, greater than, not equal to, less than or equal to, greater than or equal to).
		2.5.3	Understand the purpose of, and how to use, logic operators (AND, OR, NOT).
	Subprograms	2.6.1	understand the benefits of using subprograms and be able to write code that uses user-written and pre-existing (builtin, library) subprograms
		2.6.2	understand the concept of passing data into and out of subprograms (procedures, functions)
		2.6.3	be able to create subprograms that use parameters
Data	Binary	3.1.1	Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions.
		3.1.2	Understand how computers represent and manipulate numbers (unsigned integers, signed integers (sign and magnitude, two's complement)).
		3.1.3	Be able to convert between binary and denary whole numbers (0–255).
		3.1.4	Understand how to perform binary arithmetic (add, shifts (logical and arithmetic)) and understand the concept of overflow.
		3.1.5	Understand why hexadecimal notation is used and be able to convert between hexadecimal and binary
	Data representation	3.2.1	Understand how computers encode characters using ASCII and Unicode.
		3.2.2	Understand how bitmap images are represented in binary (pixels, resolution, colour depth).
		3.2.3	Understand how sound, an analogue signal, is represented in binary.
		3.2.4	Understand the limitations of binary representation of data (sampling frequency, resolution) when constructed by the number of available bits.

Topic	Sub Topic	International GCSE Reference	International GCSE Learning Objective
	Data storage and compression	3.3.1	Understand how to use and convert between binary and denary multiples (as defined by the International Electrotechnical Commission (IEC)): bit, nibble, byte, kibibyte (KiB) 2^10, mebibyte (MiB) 2^20, gibibyte (GiB) 2^30, tebibyte (TiB) 2^40, kilobyte (kB), 10^3, megabyte (MB) 10^6, gigabyte (GB) 10^9, terabyte (TB) 10^12
		3.3.2	Understand the need for data compression and methods of compressing data (lossless, lossy), and that JPEG and MP3 are examples of lossy algorithms.
		3.3.3	Understand how a lossless, run-length encoding (RLE) algorithm works
		3.3.4	Understand that file storage is measured in bytes and be able to calculate file sizes.
	Encryption	3.4.1	understand the need for data encryption
		3.4.2	Understand how encryption algorithms work (Pigpen cipher, Caesar cipher, Vigenére cipher, Rail Fence cipher).
Computers	Machines and computational models	4.1.1	Understand the input-process-output model.
		4.1.2	Understand that there is a range of computational models (sequential, parallel, multi-agent).
	Hardware	4.2.1	Understand the function of the hardware components of a computer system (central processing unit (CPU), main memory, secondary storage, input and output devices) and how they work together.
		4.2.2	Understand the function of different types of memory (random-access memory (RAM), read-only memory (ROM), cache, virtual memory).
		4.2.3	Understand the concept of a stored program and the role of components of the CPU (control unit (CU), arithmetic/logic unit (ALU), registers, clock, address bus, data bus, control bus) in the fetch-decodeexecute cycle (the Von Neumann model).
		4.2.4	Understand factors that affect the performance of the CPU (clock speed, number of processor cores, size of cache, type of cache).
		4.2.5	Understand how data is stored on physical devices (magnetic, optical, solid state).
		4.2.6	Understand the concept of storing data in the 'cloud' and other contemporary secondary storage.
		4.2.7	Understand the need for embedded systems and their functions.
	Logic	4.3.1	be able to construct truth tables for a given logic statement (AND, OR, NOT)
		4.3.2	be able to produce logic statements for a given problem
		4.4.1	Know what an operating system is and how it manages files, processes, hardware and the user interface.

Topic	Sub Topic	International GCSE Reference	International GCSE Learning Objective
	Software	4.4.2	Understand the purpose and functions of utility software (managing, repairing and converting files; compression; defragmentation; backing up; anti-malware (antivirus, anti-spyware)).
		4.4.3	Understand how software can be used to simulate and model aspects of the real world.
	Programming languages	4.5.1	Understand what is meant by high-level and low-level programming languages and understand their suitability for a particular task.
		4.5.2	Understand what is meant by an assembler, a compiler and an interpreter when translating programming languages and know the advantages and disadvantages of each.
Communication and the Internet	Networks	5.1.1	understand why computers are connected in a network
		5.1.2	Understand the different types of networks (local area network (LAN), wide area network (WAN), personal area network (PAN)) and usage models (client-server, peer-to-peer).
		5.1.3	understand wired and wireless connectivity
		5.1.4	understand that network data speeds are measured in bits per second (Mbps, Gbps)
		5.1.5	understand the role of and need for network protocols (Ethernet, Wi-Fi, TCP/IP, HTTP. HTTPS, FTP, email (POP3, SMTP, IMAP))
		5.1.6	Understand that data can be transmitted in packets using layered protocol stacks and the 4-layer TCP/IP model (application, transport, network, data link).
		5.1.7	understand characteristics of network topologies (bus, ring, star, mesh)
		5.1.8	Understand the different mobile communication standards (3G, 4G and subsequent generations).
	Network security	5.2.1	understand the importance of network security and be able to use appropriate validation and authentication techniques (access control, physical security and firewalls)
		5.2.2	understand security issues associated with the 'cloud' and other contemporary storage
		5.2.3	Understand different forms of cyber attack (based on technical weaknesses and behaviour), including social engineering (phishing, shoulder surfing, pharming), unpatched software, USB devices, digital devices and eavesdropping.
		5.2.4	understand methods of identifying vulnerabilities including penetration testing, ethical hacking, commercial analysis tools and review of network and user policies
		5.2.5	Understand how to protect software systems from cyber attacks, including considerations at the software (application) design stage, audit trails, securing operating systems, secure coding, code reviews to remove code vulnerabilities in programming languages and bad programming practices, modular testing and effective network security provision.

Topic	Sub Topic	International GCSE Reference	International GCSE Learning Objective
	The internet and the world wide web	5.3.1	Understand what is meant by the internet and how the internet is structured (IP addressing, domain name service (DNS)).
		5.3.2	understand what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML)
		5.3.3	Understand the need for IP addressing standards and the formats of IPv4 and IPv6.
		5.3.4	Understand the role of components used to access the internet (modem, router, switch, wireless access point (WAP)) and how these are combined.
The bigger picture	Emerging trends, issues and impact	6.1.1	understand the environmental impact of technology (health, energy use, resources) on society
		6.1.2	understand the ethical impact of using technology (privacy, inclusion, professionalism) on society
		6.1.3	Understand the legal impact of using technology (intellectual property, patents, licensing and cyber-security).
		6.1.4	Be aware of current and emerging trends in computing technology (quantum computing, DNA computing, artificial intelligence (AI), nanotechnology).

Section	Topic		CIE	Computer Science 0478 for exams in 2026, 2027 and 2028		Edexcel iG Reference	Edexcel iG Learning Objective
Computer Systems	Topic 1: Data representation	Number systems	1.1.1	Understand how and why computers use binary to represent all forms of data	3.1.1	3.1.1	Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions.
			1.1.2	(a) Understand the denary, binary and hexadecimal number systems (b) Convert between (i) positive denary and positive binary (ii) positive denary and positive hexadecimal (iii) positive hexadecimal and positive binary	3.1.3	3.1.3	Be able to convert between binary and denary whole numbers (0–255).
			1.1.3	Understand how and why hexadecimal is used as a beneficial method of data representation	3.1.5	3.1.5	Understand why hexadecimal notation is used and be able to convert between hexadecimal and binary
			1.1.4	(a) Add two positive 8-bit binary integers (b) Understand the concept of overflow and why it occurs in binary addition	3.1.4	3.1.4	Understand how to perform binary arithmetic (add, shifts (logical and arithmetic)) and understand the concept of overflow.
			1.1.5	Perform a logical binary shift on a positive 8-bit binary integer and understand the effect this has on the positive binary integer	3.1.4	3.1.4	Understand how to perform binary arithmetic (add, shifts (logical and arithmetic)) and understand the concept of overflow.
			1.1.6	Use the two's complement number system to represent positive and negative 8-bit binary integers	3.1.2	3.1.2	Understand how computers represent and manipulate numbers (unsigned integers, signed integers (sign and magnitude, two's complement)).
		Text sound and images	1.2.1	Understand how and why a computer represents text and the use of character sets, including American standard code for information interchange (ASCII) and Unicode	3.2.1	3.2.1	Understand how computers encode characters using ASCII and Unicode.
			1.2.2	Understand how and why a computer represents sound, including the effects of the sample rate and sample resolution	3.2.3	3.2.3	Understand how sound, an analogue signal, is represented in binary.
			1.2.3	Understand how and why a computer represents an image, including the effects of the resolution and colour depth	3.2.2	3.2.2	Understand how bitmap images are represented in binary (pixels, resolution, colour depth).
		Data storage & compression	1.3.1	Understand how data storage is measured Including: – bit, nibble, byte, kibibyte (KiB), mebibyte (MiB), gibibyte (GiB), tebibyte (TiB), pebibyte (PiB), exbibyte (EiB)	3.3.1	3.3.1	Understand how to use and convert between binary and denary multiples (as defined by the International Electrotechnical Commission (IEC)): bit, nibble, byte, kibibyte (KiB) 2^10, mebibyte (MiB) 2^20, gibibyte (GiB) 2^30, tebibyte (TiB) 2^40, kilobyte (kB), 10^3, megabyte (MB) 10^6, gigabyte (GB) 10^9, terabyte (TB) 10^12
			1.3.2	Calculate the file size of an image file and a sound file, using information given NB.Answers must be given in the units specified in the question. Calculations must use the measurement of 1024 and not 1000	3.3.4	3.3.4	Understand that file storage is measured in bytes and be able to calculate file sizes.
			1.3.3	Understand the purpose of and need for data compression	3.3.2	3.3.2	Understand the need for data compression and methods of compressing data (lossless, lossy), and that JPEG and MP3 are examples of lossy algorithms.
			1.3.4	Understand how files are compressed using lossy and lossless compression methods	3.3.2	3.3.2	Understand the need for data compression and methods of compressing data (lossless, lossy), and that JPEG and MP3 are examples of lossy algorithms.
Computer Systems	Topic 2: Data transmission	Types & methods of data transmission	2.1.1	(a) Understand that data is broken down into packets to be transmitted (b) Describe the structure of a packet (c) Describe the process of packet switching	5.1.6	5.1.6	Understand that data can be transmitted in packets using layered protocol stacks and the 4-layer TCP/IP model (application, transport, network, data link).
			2.1.2	(a) Describe how data is transmitted from one device to another using different methods of data transmission (b) Explain the suitability of each method of data transmission, for a given scenario	5.1.5	5.1.5	understand the role of and need for network protocols (Ethernet, Wi-Fi, TCP/IP, HTTP. HTTPS, FTP, email (POP3, SMTP, IMAP))
			2.1.3	Understand the universal serial bus (USB) interface and explain how it is used to transmit data			Not in Edexcel Specification
		Methods of error detection	2.2.1	Understand the need to check for errors after data transmission and how these errors can occur			Not in Edexcel Specification
			2.2.2	Describe the processes involved in each of the following error detection methods for detecting errors in data after transmission: parity check (odd and even), checksum and echo check			Not in Edexcel Specification
			2.2.3	Describe how a check digit is used to detect errors in data entry and identify examples of when a check digit is used, including international standard book numbers (ISBN) and bar codes			Not in Edexcel Specification
			2.2.4	Describe how an automatic repeat query (ARQ) can be used to establish that data is received without error			Not in Edexcel Specification
		Encryption	2.3.1	Understand the need for and purpose of encryption when transmitting data	3.4.1	3.4.1	understand the need for data encryption
			2.3.2	Understand how data is encrypted using symmetric and asymmetric encryption	3.4.2	3.4.2	Understand how encryption algorithms work (Pigpen cipher, Caesar cipher, Vigenére cipher, Rail Fence cipher).
Computer Systems	Hardware	Computer archetecture	3.1.1	(a) Understand the role of the central processing unit (CPU) in a computer (b) Understand what is meant by a microprocessor	4.2.1	4.2.1	Understand the function of the hardware components of a computer system (central processing unit (CPU), main memory, secondary storage, input and output devices) and how they work together.
			3.1.2	(a) Understand the purpose of the components in a CPU, in a computer that has a Von Neumann architecture (b) Describe the process of the fetch–decode–execute (FDE) cycle, including the role of each component in the process	4.2.3	4.2.3	Understand the concept of a stored program and the role of components of the CPU (control unit (CU), arithmetic/logic unit (ALU), registers, clock, address bus, data bus, control bus) in the fetch-decodeexecute cycle (the Von Neumann model).
			3.1.3	Understand what is meant by a core, cache and clock in a CPU and explain how they can affect the performance of a CPU	4.2.4	4.2.4	Understand factors that affect the performance of the CPU (clock speed, number of processor cores, size of cache, type of cache).
			3.1.4	Understand the purpose and use of an instruction set for a CPU	4.2.3	4.2.3	Understand the concept of a stored program and the role of components of the CPU (control unit (CU), arithmetic/logic unit (ALU), registers, clock, address bus, data bus, control bus) in the fetch-decodeexecute cycle (the Von Neumann model).
			3.1.5	Describe the purpose and characteristics of an embedded system and identify devices in which they are commonly used	4.2.7	4.2.7	Understand the need for embedded systems and their functions.
		Input & output devices	3.2.1	Understand what is meant by an input device and why it is required			Not in Edexcel Specification
			3.2.2	Understand what is meant by an output device and why it is required			Not in Edexcel Specification
			3.2.3	(a) Understand what is meant by a sensor and the purposes of sensors (b) Identify the type of data captured by each sensor and understand when each sensor would be used, including selecting the most suitable sensor for a given context			Not in Edexcel Specification

Section	Topic		CIE	Computer Science 0478 for exams in 2026, 2027 and 2028		Edexcel iG Reference	Edexcel iG Learning Objective
Computer Systems	Topic 3:	Data storage	3.3.1	Understand what is meant by primary storage	4.2.2	4.2.2	Understand the function of different types of memory (random-access memory (RAM), read-only memory (ROM), cache, virtual memory).
			3.3.2	Understand what is meant by secondary storage	4.2.2	4.2.2	Understand the function of different types of memory (random-access memory (RAM), read-only memory (ROM), cache, virtual memory).
			3.3.3	Describe the operation of magnetic, optical and solid-state (flash memory) storage and give examples of each	4.2.5	4.2.5	Understand how data is stored on physical devices (magnetic, optical, solid state).
			3.3.4	Describe what is meant by virtual memory, how it is created and used and why it is necessary	4.2.2	4.2.2	Understand the function of different types of memory (random-access memory (RAM), read-only memory (ROM), cache, virtual memory).
			3.3.5	Understand what is meant by cloud storage	4.2.6	4.2.6	Understand the concept of storing data in the 'cloud' and other contemporary secondary storage.
			3.3.6	Explain the advantages and disadvantages of storing data on the cloud in comparison to storing it locally	5.2.2	5.2.2	understand security issues associated with the 'cloud' and other contemporary storage
		Network hardware	3.4.1	Understand that a computer needs a network interface card (NIC) to access a network			Not in Edexcel Specification
			3.4.2	Understand what is meant by, and the purpose of, a media access control (MAC) address, including its structure			Not in Edexcel Specification
			3.4.3	(a) Understand what is meant by, and the purpose of, an internet protocol (IP) address (b) Understand that there are different types of IP address	5.3.1	5.3.1	Understand what is meant by the internet and how the internet is structured (IP addressing, domain name service (DNS)).
			3.4.4	Describe the role of a router in a network	5.3.4	5.3.4	Understand the role of components used to access the internet (modem, router, switch, wireless access point (WAP)) and how these are combined.
	Topic 4: Software	Types of software and interrupts	4.1.1	Describe the difference between system software and application software and provide examples of each	4.4.2	4.4.2	Understand the purpose and functions of utility software (managing, repairing and converting files; compression; defragmentation; backing up; anti-malware (antivirus, anti-spyware)).
			4.1.2	Describe the role and basic functions of an operating system	4.4.1	4.4.1	Know what an operating system is and how it manages files, processes, hardware and the user interface.
			4.1.3	Understand how hardware, firmware and an operating system are required to run applications software	4.4.2	4.4.2	Understand the purpose and functions of utility software (managing, repairing and converting files; compression; defragmentation; backing up; anti-malware (antivirus, anti-spyware)).
			4.1.4	Describe the role and operation of interrupts			Not in Edexcel Specification
		Types of programming language, translators and integrated development environments (IDEs)	4.2.1	Explain what is meant by a high-level language and a low-level language, including the advantages and disadvantages of each	4.5.1	4.5.1	Understand what is meant by high-level and low-level programming languages and understand their suitability for a particular task.
			4.2.2	Understand that assembly language is a form of low-level language that uses mnemonics, and that an assembler is needed to translate an assembly language program into machine code	4.5.1	4.5.1	Understand what is meant by high-level and low-level programming languages and understand their suitability for a particular task.
			4.2.3	Describe the operation of a compiler and an interpreter, including how high-level language is translated by each and how errors are reported	4.5.2	4.5.2	Understand what is meant by an assembler, a compiler and an interpreter when translating programming languages and know the advantages and disadvantages of each.
			4.2.4	Explain the advantages and disadvantages of a compiler and an interpreter	4.5.2	4.5.2	Understand what is meant by an assembler, a compiler and an interpreter when translating programming languages and know the advantages and disadvantages of each.
			4.2.5	Explain the role of an IDE in writing program code and the common functions IDEs provide			Not in Edexcel Specification
Computer Systems	Topic 5: The internet and its uses	The internet and the world wide web	5.1.1	Understand the difference between the internet and the world wide web	5.3.2	5.3.2	understand what is meant by the world wide web (WWW) and components of the WWW (web server URLs,
			5.1.2	Understand what is meant by a uniform resource locator (URL)	5.3.2	5.3.2	understand what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML)
			5.1.3	Describe the purpose and operation of hypertext transfer protocol (HTTP) and hypertext transfer protocol secure (HTTPS)	5.3.2	5.3.2	understand what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML)
			5.1.4	Explain the purpose and functions of a web browser			Not in Edexcel Specification
			5.1.5	Describe how web pages are located, retrieved and displayed on a device when a user enters a URL	5.1.5	5.1.5	understand the role of and need for network protocols (Ethernet, Wi-Fi, TCP/IP, HTTP. HTTPS, FTP, email (POP3, SMTP, IMAP))
			5.1.6	Explain what is meant by cookies and how they are used, including session cookies and persistent cookies			Not in Edexcel Specification
		Digital currency	5.2.1	Understand the concept of a digital currency and how digital currencies are used			Not in Edexcel Specification
			5.2.2	Understand the process of blockchain and how it is used to track digital currency transactions			Not in Edexcel Specification
		Cyber security	5.3.1	Describe the processes involved in, and the aim of carrying out, a range of cyber security threats	5.2.3	5.2.3	Understand different forms of cyber attack (based on technical weaknesses and behaviour), including social engineering (phishing, shoulder surfing, pharming), unpatched software, USB devices, digital devices and eavesdropping.
					5.2.4	5.2.4	understand methods of identifying vulnerabilities including penetration testing, ethical hacking, commercial analysis tools and review of network and user policies
			5.3.2	Explain how a range of solutions are used to help keep data safe from security threats	5.2.5	5.2.5	Understand how to protect software systems from cyber attacks, including considerations at the software (application) design stage, audit trails, securing operating systems, secure coding, code reviews to remove
	Topic 6: Automated and emerging technologies	Automated systems	6.1.1	Describe how sensors, microprocessors and actuators can be used in collaboration to create automated systems			Not in Edexcel Specification
			6.1.2	Describe the advantages and disadvantages of an automated system used for a given scenario			Not in Edexcel Specification
		Robotics	6.2.1	Understand what is meant by robotics			Not in Edexcel Specification
			6.2.2	Describe the characteristics of a robot			Not in Edexcel Specification
			6.2.3	Understand the roles that robots can perform and describe the advantages and disadvantages of their use			Not in Edexcel Specification
		Artificial Intelligence	6.3.1	Understand what is meant by artificial intelligence (AI)	6.1.4	6.1.4	Be aware of current and emerging trends in computing technology (quantum computing, DNA computing, artificial intelligence (AI), nanotechnology).
			6.3.2	Describe the main characteristics of AI as the collection of data and the rules for using that data, the ability to reason, and it can include the ability to learn and adapt			Not in Edexcel Specification

Section	Topic		CIE	Computer Science 0478 for exams in 2026, 2027 and 2028		Edexcel iG Reference	Edexcel iG Learning Objective
		AI int	6.3.3	Explain the basic operation and components of AI systems to simulate intelligent behaviour			Not in Edexcel Specification
Algorithms, programming and logic	Topic 7: Algorithm design and problem-solving		7.1	Understand the program development life cycle, limited to: analysis, design, coding and testing			Not in Edexcel Specification
			7.2	(a) Understand that every computer system is made up of sub-systems, which are made up of further sub-systems (b) Understand how a problem can be decomposed into its component parts (c) Use different methods to design and construct a solution to a problem	4.1.1 1.2.2	4.1.1 1.2.2	Understand the input-process-output model. Be able to decompose a problem into smaller sub-problems.
			7.3	Explain the purpose of a given algorithm	1.1.1	1.1.1	Understand what an algorithm is, what algorithms are used for and be able to interpret algorithms (flowcharts, pseudocode, written descriptions, program code).
			7.4	Understand standard methods of solution			Not in Edexcel Specification
			7.5	(a) Understand the need for validation checks to be made on input data and the different types of validation check (b) Understand the need for verification checks to be made on input data and the different types of verification check	2.4.2 2.1.4	2.4.2 2.1.4	Understand the need for, and how to implement, validation Be able to design and use test plans and test data (normal, boundary, erroneous).
			7.6	Suggest and apply suitable test data	2.1.4	2.1.4	Be able to design and use test plans and test data (normal, boundary, erroneous).
			7.7	Complete a trace table to document a dry-run of an algorithm	2.1.6	2.1.6	Be able to determine what value a variable will hold at a given point in a program (trace table).
			7.8	Identify errors in given algorithms and suggest ways of correcting these errors	2.1.5	2.1.5	Be able to interpret error messages and identify, locate and fix errors in a program.
			7.9	Write and amend algorithms for given problems or scenarios, using: pseudocode, program code and flowcharts	1.1.2 2.1.1	1.1.2 2.1.1	Understand how to create an algorithm to solve a particular problem, making use of programming constructs (sequence, selection, iteration) and using appropriate conventions (flowchart, pseudocode, written description, draft program code). Be able to program abstractions of real-world examples.
	Topic 8: Programming	Programming concepts	8.1.1	Declare and use variables and constants	2.2.1	2.2.1	Understand the structural components of a program (variable and type declarations, command sequences, selection, iteration, data structures, subprograms).
			8.1.2	Understand and use basic data types Including: integer, real, char, string, Boolean	2.3.1	2.3.1	Understand the need for, and understand how to use, data types (integer, real, Boolean, char, string).
			8.1.3	Understand and use input and output	2.4.3	2.4.3	Be able to write code that reads/writes from/to a text file.
			8.1.4	(a) Understand and use the concept of sequence (b) Understand and use the concept of selection (c) Understand and use the concept of iteration (d) Understand and use the concepts of totalling and counting (e) Understand and use the concept of string handling (f) Understand and use arithmetic, logical and Boolean operators	2.2.2	2.2.2	Be able to use sequencing, selection and iteration constructs in their programs.
			8.1.5	Understand and use nested statements			Not in Edexcel Specification
			8.1.6	(a) Understand what is meant by procedures, functions and parameters (b) Define and use procedures and functions, with or without parameters (c) Understand and use local and global variables	2.6.1	2.6.1	understand the benefits of using subprograms and be able to write code that uses user-written and pre-existing (builtin, library) subprograms
			8.1.7	Understand and use library routines Including: MOD, DIV, ROUND, RANDOM	2.6.1	2.6.1	understand the benefits of using subprograms and be able to write code that uses user-written and pre-existing (builtin, library) subprograms
			8.1.8	Understand how to create a maintainable program	2.1.2	2.1.2	Understand the benefit of producing programs that are easy to read and be able to use techniques (comments, descriptive names (variables, constants, subprograms), indentation) to improve readability and to explain how the code works.
		Arrays	8.2.1	Declare and use one-dimensional (1D) and two-dimensional (2D) arrays	2.3.2	2.3.2	Understand the need for, and understand how to use, data structures (records, one-dimensional arrays, two-dimensional arrays).
			8.2.2	Understand the use of arrays	2.3.2	2.3.2	Understand the need for, and understand how to use, data structures (records, one-dimensional arrays, two-dimensional arrays).
			8.2.3	Write values into, and read values from, an array using iteration			Not in Edexcel Specification
		File handling	8.3.1	Understand the purpose of storing data in a file to be used by a program			Not in Edexcel Specification
			8.3.2	Open, close and use a file for reading and writing	2.4.3	2.4.3	Be able to write code that reads/writes from/to a text file.
	Topic 9: Databases		9.1	Define a single-table database from given data storage requirements			Not in Edexcel Specification
			9.2	Suggest suitable basic data types			Not in Edexcel Specification
			9.3	Understand the purpose of a primary key and identify a suitable primary key for a given database table			Not in Edexcel Specification
			9.4	Read, understand and complete structured query language (SQL) scripts to query data stored in a single database table			Not in Edexcel Specification
	olean logic		10.1	Identify and use the standard symbols for logic gates			Not in Edexcel Specification
			10.2	Define and understand the functions of logic gates • Including: NOT, AND, OR, NAND, NOR, XOR (EOR) – the binary output produced from all the possible binary inputs • NOT is a single input gate • All other gates are limited to two inputs			Not in Edexcel Specification

Section	Topic		CIE	Computer Science 0478 for exams in 2026, 2027 and 2028	Edexcel iG Reference	Edexcel iG Learning Objective
	Topic 10: Boo		10.3	(a) Use logic gates to create given logic circuits from a: (i) problem statement, (ii) logic expression, (iii) truth table (b) Complete a truth table from a: (i) problem statement, (ii) logic expression, (iii) logic circuit (c) Write a logic expression from a: (i) problem statement (ii) logic circuit (iii) truth table		Not in Edexcel Specification