

# Summer 2023 Computer Science GCSE OCR

Paper 1 (01)			PG online pages:
<b>1.1 Systems Architecture</b>	<b>1.1.1 Architecture of the CPU</b>	<ul style="list-style-type: none"> <li>The purpose of the CPU                             <ul style="list-style-type: none"> <li>- What actions occur at each stage of the fetch-execute cycle</li> <li>- The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle</li> </ul> </li> <li>Common CPU components and their function:                             <ul style="list-style-type: none"> <li>-ALU (Arithmetic Logic Unit), CU (Control Unit), cache, registers</li> </ul> </li> <li>Von Neumann architecture                             <ul style="list-style-type: none"> <li>-MAR (Memory Address Register), MDR (Memory Data Register), program counter, accumulator.</li> </ul> </li> </ul>	2-3
	<b>1.1.2 CPU performance</b>	<ul style="list-style-type: none"> <li>How common characteristics of CPU's affect their performance:                             <ul style="list-style-type: none"> <li>-Clock speed</li> <li>-Cache size</li> <li>-Number of cores</li> </ul> </li> </ul>	4
	<b>1.1.3 Embedded systems</b>	<ul style="list-style-type: none"> <li>The purpose and characteristics of embedded systems</li> <li>Examples of embedded systems.</li> </ul>	4-5
<b>1.2 Memory and storage</b>	<b>1.2.1 Primary storage (Memory)</b>	<ul style="list-style-type: none"> <li>The need for primary storage.</li> <li>The difference between RAM and ROM.</li> <li>The purpose of ROM in a computer system.</li> <li>The purpose of RAM in a computer system.</li> <li>Virtual memory- The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle.</li> </ul>	6-7
	<b>1.2.2 Secondary storage</b>	<ul style="list-style-type: none"> <li>The need for secondary storage</li> <li>Common types of storage:                             <ul style="list-style-type: none"> <li>-optical, magnetic, solid state</li> </ul> </li> <li>Suitable storage devices and storage media for a given application.</li> <li>The advantages and disadvantages of different storage devices and storage media relating to these characteristics:                             <ul style="list-style-type: none"> <li>-capacity</li> <li>-speed</li> <li>-portability</li> <li>-durability</li> <li>-reliability</li> <li>-cost</li> </ul> </li> </ul>	8,9
	<b>1.2.3 Units</b>	Why data must be stored in binary format. <ul style="list-style-type: none"> <li>Calculate required storage capacity for a given set of files.</li> <li>Calculate file sizes of sound, images and text files.</li> <li>Sound file size = sample rate x duration (s) x bit depth    image file size = colour depth x image height (px) x image width (px)</li> <li>text file size = bits per character x number of characters.</li> </ul>	11
	<b>1.2.4 Data Storage</b>	Numbers: <ul style="list-style-type: none"> <li>How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa.</li> <li>How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur.</li> <li>How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa.</li> <li>How to convert binary integers to their hexadecimal equivalents and vice versa</li> </ul>	12- 20

		<ul style="list-style-type: none"> <li>• Binary shifts.</li> </ul> <p>Characters:</p> <ul style="list-style-type: none"> <li>• The use of binary codes to represent characters.</li> <li>• The term ‘character set’.</li> <li>• The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ASCII, Unicode.</li> </ul> <p>Images:</p> <ul style="list-style-type: none"> <li>• How an image is represented as a series of pixels, represented in binary.</li> <li>• Metadata.</li> <li>• The effect of colour depth and resolution on: The quality of the image, the size of an image file.</li> </ul> <p>Sound:</p> <ul style="list-style-type: none"> <li>• How sound can be sampled and stored in digital form.</li> <li>• The effect of sample rate, duration and bit depth on: The playback quality, The size of a sound file.</li> </ul>	
<b>1.3 Computer Networks, connections and protocols</b>	<b>1.2.5 Compression</b>	<ul style="list-style-type: none"> <li>• The need for compression.</li> <li>• Types of compression: Lossy, Lossless.</li> <li>• Advantages and disadvantages of each type of compression</li> <li>• Effects on the file for each type of compression.</li> </ul>	21
	<b>1.3.1 Networks and topologies</b>	<ul style="list-style-type: none"> <li>• Types of networks: LAN (Local Area Network), WAN (Wide Area Network)</li> <li>• Factors that affect the performance of networks.</li> <li>• The different roles of computers in a client-server and a peer-to-peer network.</li> <li>• The hardware needed to connect stand-alone computers into a Local Area Network: Wireless access points, Routers, Switches, NIC (Network Interface Controller/Card), Transmission media.</li> <li>• The Internet as a worldwide collection of computer network s: DNS (Domain Name Server), Hosting, The Cloud, Web servers and clients</li> <li>• Star and Mesh network topologies- advantages/disadvantages.</li> <li>• Modes of connection: Wired (Ethernet), Wireless (Wi-Fi, Bluetooth)</li> <li>• Encryption</li> <li>• IP addressing and MAC addressing</li> </ul>	23-29
	<b>1.3.2 Wired and wireless networks, protocols and layers</b>	<ul style="list-style-type: none"> <li>• Modes of connection: Wired (Ethernet), Wireless (Wi-Fi, Bluetooth)</li> <li>• Encryption</li> <li>• IP addressing and MAC addressing</li> <li>• The principle of a standard to provide rules for areas of computing.</li> <li>• Common protocols including: TCP/IP (Transmission Control Protocol/Internet Protocol), HTTP (Hyper Text Transfer Protocol) , HTTPS (Hyper Text Transfer Protocol Secure) , FTP (File Transfer Protocol), POP (Post Office Protocol) ,, IMAP (Internet Message Access Protocol, SMTP (Simple Mail Transfer Protocol)</li> <li>• 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.</li> </ul>	30-32
<b>1.4 Network Security</b>	<b>1.4.1 Threats to computer systems and networks</b>	<ul style="list-style-type: none"> <li>• Forms of attack: Malware, Social engineering, e.g., phishing, people as the ‘weak point’, Brute-force attacks, Denial of service attacks, Data interception and theft, The concept of SQL injection.</li> <li>• Common prevention methods: Penetration testing, Anti-malware software, Firewalls, User access levels, Passwords, Encryption, Physical security.</li> </ul>	34

	<b>1.4. Identifying and preventing vulnerabilities</b>	<ul style="list-style-type: none"> <li>Knowledge/principles of each prevention method: What each prevention method may limit/prevent, how it limits the attack</li> </ul>	35
<b>1.5 Systems software</b>	<b>1.5.1 Operating systems</b>	<ul style="list-style-type: none"> <li>What each function of an operating system does.</li> <li>Features of a user interface.</li> <li>Memory management, e.g., the transfer of data between memory, and how this allows for multitasking.</li> <li>Understand that: Data is transferred between devices and the processor; this process needs to be managed.</li> <li>User management functions, e.g.: Allocation of an account, Access rights, Security, etc.</li> <li>What each function of an operating system does. Features of a user interface.</li> <li>Memory management, e.g., the transfer of data between memory, and how this allows for multitasking.</li> <li>Understand that: Data is transferred between devices and the processor; this process needs to be managed.</li> <li>User management functions, e.g.: Allocation of an account, Access rights, Security, etc.</li> <li>File management, and the key features, e.g.: Naming &amp; Allocating to folders, moving files, Saving, etc.</li> </ul>	37
	<b>1.5.2 Utility software</b>	<ul style="list-style-type: none"> <li>The purpose and functionality of utility software.</li> <li>Utility system software: Encryption software, Defragmentation, Data compression.</li> </ul>	38
<b>1.6 Ethical, legal, cultural and environmental impacts of digital technology</b>	<b>1.6.1 Ethical, legal, cultural and environmental impact</b>	<ul style="list-style-type: none"> <li>Impacts of digital technology on wider society including Ethical issues, Legal issues, Cultural issues, Environmental issues, Privacy issues.</li> <li>Legislation relevant to Computer Science: The Data Protection Act 2018, Computer Misuse Act 1990, Copyright Designs and Patents Act 1988, Software licences (i.e., open source and proprietary).</li> <li>Features of open source (providing access to the source code and the ability to change the software).</li> <li>Features of proprietary (no access to the source code, purchased commonly as off-the-shelf).</li> <li>Recommend a type of licence for a given scenario including benefits and drawbacks.</li> </ul>	40-44

<b>Paper 2 (02)</b>			
			<b>PG online pages:</b>
<b>2.1 Algorithms</b>	<b>2.1.1 Computational thinking</b>	<ul style="list-style-type: none"> <li>Abstraction, Decomposition, Algorithmic thinking.</li> </ul>	47
	<b>2.1.2 Designing, creating and refining algorithms</b>	<ul style="list-style-type: none"> <li>Identify the inputs, processes, and outputs for a problem.</li> <li>Structure diagrams.</li> <li>Create, interpret, correct, complete, and refine algorithms using: Flowcharts, Reference language/high-level programming language.</li> </ul>	48-53

		<ul style="list-style-type: none"> <li>Identify common errors.</li> <li>Trace tables. (How to use them, how many rows would be needed)</li> </ul>	
2.2 Programming fundamentals	2.1.3 Searching and sorting algorithms	<ul style="list-style-type: none"> <li>Standard searching algorithms: Binary search, Linear search.</li> <li>Standard sorting algorithms: Bubble sort, Merge sort, Insertion sort.</li> </ul>	54-60
	2.2.1 Programming fundamentals	<ul style="list-style-type: none"> <li>The use of variables, constants, operators, inputs, outputs and assignments.</li> <li>The use of the three basic programming constructs used to control the flow of a program: Sequence, Selection, Iteration (count- and condition-controlled loops).</li> <li>The common arithmetic operators.</li> <li>The common Boolean operators AND, OR and NOT.</li> </ul>	61-77
	2.2.2 Data Types	<ul style="list-style-type: none"> <li>The use of data types: Integer, Real, Boolean, Character and string, Casting.</li> </ul>	
2.3 Producing robust programs	2.2.3 Additional programming techniques	<ul style="list-style-type: none"> <li>The use of basic string manipulation.</li> <li>The use of basic file handling operations: Open, Read, Write, Close.</li> <li>The use of records to store data.</li> <li>The use of SQL to search for data. (SELECT, FROM, WHERE).</li> <li>The use of arrays (or equivalent) when solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D).</li> <li>How to use sub programs (functions and procedures) to produce structured code.</li> <li>Random number generator.</li> </ul>	
	2.3.1 Defensive design	<ul style="list-style-type: none"> <li>Defensive design considerations: Anticipating misuse, Authentication.</li> <li>Input validation – how to deal with invalid data.</li> <li>Maintainability: Use of sub programs, Naming conventions, Indentation, Commenting.</li> </ul>	78
	2.3.2 Testing	<ul style="list-style-type: none"> <li>The difference between testing modules of a program during development and testing the program at the end of production.</li> <li>-Types of testing: Iterative, Final/terminal</li> <li>Identify syntax and logic errors</li> <li>Selecting and using suitable test data: Normal, Boundary, Invalid/Erroneous</li> <li>Refining algorithms-making them more effective</li> </ul>	80
2.4 Boolean logic	2.4.1. Boolean logic	<ul style="list-style-type: none"> <li>Simple logic diagrams using the operators AND, OR and NOT.</li> <li>Knowledge of Truth tables for each logic gate.</li> <li>Combining Boolean operators using AND, OR and NOT</li> <li>Applying logical operators in truth tables to solve problems.</li> </ul>	82
2.5 Programming languages and integrated development environments	2.5.1 Languages	<ul style="list-style-type: none"> <li>Characteristics and purpose of different levels of programming language: High-level language, Low-level languages.</li> <li>The purpose of translators, need for translators.</li> <li>The characteristics of a compiler and an interpreter. The differences, benefits and drawbacks of using a compiler or an interpreter.</li> </ul>	84
	2.5.2 The integrated development environment (IDE)	<ul style="list-style-type: none"> <li>Common tools and facilities available in an Integrated Development Environment (IDE): Editors, Error diagnostics, Run-time environment, Translators.</li> <li>How each of the tools and facilities listed can be used to help a programmer develop a program.</li> </ul>	85

