GCE A LEVEL MARKING SCHEME

SUMMER 2024

**A LEVEL
COMPUTER SCIENCE - UNIT 3
1500U30-1**

About this marking scheme

The purpose of this marking scheme is to provide teachers, learners, and other interested parties, with an understanding of the assessment criteria used to assess this specific assessment.

This marking scheme reflects the criteria by which this assessment was marked in a live series and was finalised following detailed discussion at an examiners' conference. A team of qualified examiners were trained specifically in the application of this marking scheme. The aim of the conference was to ensure that the marking scheme was interpreted and applied in the same way by all examiners. It may not be possible, or appropriate, to capture every variation that a candidate may present in their responses within this marking scheme. However, during the training conference, examiners were guided in using their professional judgement to credit alternative valid responses as instructed by the document, and through reviewing exemplar responses.

Without the benefit of participation in the examiners' conference, teachers, learners and other users, may have different views on certain matters of detail or interpretation. Therefore, it is strongly recommended that this marking scheme is used alongside other guidance, such as published exemplar materials or Guidance for Teaching. This marking scheme is final and will not be changed, unless in the event that a clear error is identified, as it reflects the criteria used to assess candidate responses during the live series.

GCE A LEVEL COMPUTER SCIENCE - UNIT 3

SUMMER 2024 MARK SCHEME

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
1 (a)	**Award 1 mark for edit for 2D and 3D maximum of 2 marks.**					4
	A two-dimensional array is a collection of values stored in rows and columns.	1				
	A two-dimensional array can be thought of as a table of values or a matrix.	1				
	Each value in a two-dimensional array can be accessed using two indices e.g. 2D [1][1]	1				
	A three-dimensional array is a collection of values stored in rows, columns and layers.	1	1b			
	A three-dimensional array can be thought of as a cube of values or a tensor.	1				
	Each value in a three-dimensional array can be accessed using three indices e.g. 3D [1][1][1]	1				
	Award 1 mark for each correct example up to a maximum of two marks.					
	An example use of a two-dimensional array would be to store a table of player scores in a game.	1				
	An example use of a three-dimensional array would be to store the RGB colour channel values of a specific pixel in an image.	1				
	Accept any suitable example(s).					
1 (b)	The value can be updated by setting `july[5,6] = "X"`	1				
	Award 1 mark for correct array name july and assignment of string `"X"` **Award 1 mark for correct indices** `[6,5]`/`[5,6]`	1		2a		2
	Accept `[6][5]`					

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
1 (c)	The value can be updated by setting 2D[2,1] = 10 **Award 1 mark for correct array** 2D **and assignment of integer** 10 **Award 1 mark for correct indices** [2,1] **Accept** [2][1]	1 1		2a		2
2 (a)	**Award 1 mark for each correct response up to a maximum of 4 marks.** Standardisation of computer languages ensures that different systems can understand and communicate with each other using the same language. Standardisation helps to avoid bugs and errors that can occur when different systems use the same language in different ways. Standardisation enables interoperability, making it easier to develop software that works across different platforms and devices. Standardisation makes the software development process simpler by providing a clear set of rules for writing and understanding code. Standardisation makes it easier for developers to learn and use a language as they only need to learn one standard rather than multiple variations. Standardisation enable developers to create secure and high-quality software across different devices. An example of a computer language standard is ECMAScript which is the syntax standard used for JavaScript. **Accept any suitable example(s).**	1 1 1 1 1 1 1	1b			4

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
2 (b)	**Award 1 mark for each correct response up to a maximum of 3 marks.**					4
	Different companies may have different goals and priorities, making it difficult to agree on what should be included in a standard.	1				
	The process of developing a standard can be time-consuming and complex, involving multiple iterations and revisions.	1				
	There may be opposition from developers who are used to working with a different, non-standard version of the language.	1				
	Existing software systems may not be compatible with new agreed standards, leading to a resistance for change from some companies.	1	1b			
	There may be a potential conflict or disagreement between competing business interests of companies.	1				
	Award 1 mark for correct example					
	An example of a computer language standard that faced difficulties is HTML5 which took several years, and multiple development iterations led by the World Wide Web Consortium (W3C).	1				
	Accept any suitable example(s).					

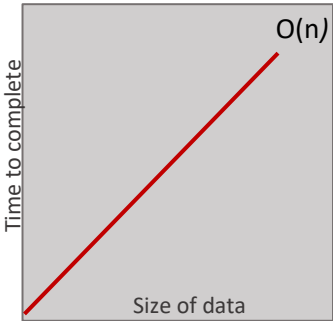| Question	Answer	Mark	AO1	AO2	AO3	TOTAL
3 (a)	Correct answer can be established using different steps / laws / rules / identities / dual relations. **Indicative content** $(1.X + \overline{Y.\overline{Y}}).\overline{X} + X.Z$ $(1.X + \overline{Y} + \overline{\overline{Y}}).\overline{X} + X.Z$ *De Morgan's* $(X + \overline{Y} + \overline{\overline{Y}}).\overline{X} + X.Z$ *Identity* $(X + \overline{Y}).\overline{X} + X.Z$ *Idempotent* $\overline{X}.X + \overline{X}.\overline{Y} + X.Z$ *Distribution* $0 + \overline{X}.\overline{Y} + X.Z$ *Complement* $\overline{X}.\overline{Y} + X.Z$ *Identity* **Award 1 mark for correctly applying De Morgan's Law.** **Award 5 marks for applying identities to arrive at correct answer.** Correctly applying identities but arriving at wrong answer 1 mark for each correct application with a max of 4 marks.	1 5		2a		6
3 (b)	Correct answer can be established using different steps / laws / rules / identities / dual relations. **Indicative content** $(\overline{A}.A + A.A).1 + B.C.0$ $(\overline{A}.A + A.A).1 + 0$ *Identity* $(\overline{A}.A + A.A).1$ *Identity* $\overline{A}.A + A.A$ *Identity* $\overline{A}.A + A$ *Idempotent* $0 + A$ *Complement* A *Complement* **Award 6 marks for correctly applying identities to arrive at correct answer.** Correctly applying identities but arriving at wrong answer 1 mark for each correct application with a max of 5 marks.	6		2a		6

4

Question	Answer	Mark	AO1	AO2	AO3	TOTAL

4	**Indicative content**
	
	``
	declare i, j, length as integer
	declare myArray as string[]
	
	set i = 0
	set length = len(myArray[])- 1
	
	for i to length
	   if myArray[i] <> "X" then
	      set j = i + 1
	      for j to length
	         if myArray[j] <> "X" then
	            if myArray[i] = myArray[j] then
	               set myArray[j] = "X"
	            end if
	         end if
	      next j
	   end if
	next i
	
	output myArray
	``
	
	Declare and initialise variables
	Correct determination of length
	Use of outer loop
	Use of inner loop
	Correct starting condition for inner loop
	Correct condition to check for duplication
	Set any duplicate values to `"X"`
	Output array
		
		1
		1
		1
		1
		1
		1
		1
		1
					3c	8

5 (a)	**Indicative content** <table><tr><th>Password</th><th>Expected Output</th><th>x</th><th>y</th><th>z</th><th>valid</th></tr><tr><td>abc</td><td>message 1</td><td>FALSE</td><td>FALSE</td><td>FALSE</td><td>FALSE</td></tr><tr><td>abc123</td><td>message 1</td><td>FALSE</td><td>TRUE</td><td>FALSE</td><td>FALSE</td></tr><tr><td>password</td><td>message 1</td><td>TRUE</td><td>FALSE</td><td>FALSE</td><td>FALSE</td></tr><tr><td>P@ssw0rd</td><td>message 2</td><td>TRUE</td><td>TRUE</td><td>TRUE</td><td>TRUE</td></tr><tr><td>pass12345</td><td>message 1</td><td>TRUE</td><td>TRUE</td><td>FALSE</td><td>FALSE</td></tr><tr><td>!@#$%^&*</td><td>message 1</td><td>TRUE</td><td>FALSE</td><td>TRUE</td><td>FALSE</td></tr><tr><td>p@ss</td><td>message 1</td><td>FALSE</td><td>FALSE</td><td>TRUE</td><td>FALSE</td></tr><tr><td>1!2!3</td><td>message 1</td><td>FALSE</td><td>TRUE</td><td>TRUE</td><td>FALSE</td></tr></table> **Award 2 marks per row to a maximum of 6 marks. One mark for test data and one mark for correct outputs**	 6			2a	6

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
5 (b)	Message 1: "Password must be at least 8 characters long and contain at least one digit and one special character." Message 2: "Password accepted." **Accept any suitable response.**	1 1		2a		2
6 (a)	The solution should be easy to use and understand for the intended users. An example is a website that has clear navigation, and an intuitive design would be highly usable. **Accept any suitable example.**	1 1	1b			2
6 (b)	The solution should be able to complete tasks within an acceptable time frame and allocated memory usage. An example is a software application that can process large data efficiently and quickly without slowing down, would be high performing. **Accept any suitable example.**	1 1	1b			2
6 (c)	The solution should be able to adapt to changing usage and be able to scale up or down as necessary. An example is an online ticket platform that can scale up and handle a sudden surge in traffic would be considered highly scalable. **Accept any suitable example.**	1 1	1b			2
6 (d)	The solution should protect data and be resilient to hacking and other security threats. An example is online banking system that uses two-factor authentication to protect users from unauthorised access would be considered highly secure. **Accept any suitable example.**	1 1	1b			2

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
7	**Indicative content** <alpha> ::= a|b|c…x|y|z|A|B|C…X|Y|Z <digit> ::= 0|1|2 . . . 8|9 <alpha-digit> ::= <alpha>|<digit> <protocol> ::= "http" | "https" <domain-char> ::= <alpha-digit> | "-" | "." <domain> ::= <domain-char> | <domain-char><domain> <path-char> ::= <alpha-digit> | <alpha-digit> <path-char> | "/"<path-char> | "/" <optional-path> ::= "/"<path-char> | "" <url>::=<protocol> "://"<domain> <optional-path> One mark awarded for correct definition of alpha and digit. One mark awarded for each correct definition up to a maximum of 5 marks. Answer not correct if BNF notation used incorrectly. Max of 5 marks.			2b		6

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
8	**Award 1 mark for each correct response up to a maximum of 6 marks.**					6
	Program version management or version control systems (VCS) is the process of managing different versions of a software program throughout its development lifecycle.	1				
	Program version management involves tracking changes made to the program's source code and managing different versions or releases of the program.	1				
	Version control systems help developers keep track of changes made to the source code over time and collaborate with other developers on the same project.	1				
	VCS can help prevent conflicts or overwriting of code and allows developers to revert to earlier versions of the code if necessary.	1				
	Program version management typically involves creating a repository for the program's source code, where all changes and versions are stored such as GitHub.	1				
	Developers can make changes to the code in their own working copies and then commit those changes back to a repository.	1	1b			
	The version control system keeps track of the changes and allows developers to view the entire history of the codebase.	1				
	Program version management also allows for the creation of different branches or forks of the code.	1				
	These branches can then be merged back into the main codebase once they are ready.	1				
	Program version management helps to ensure that code is organised and changes are tracked and managed.	1				
	Local version control stores all the changes on the local computer, whereas centralised version control has a single central repository, each user then gets their version, but only the centralised copy gets updated.	1				
	An example of a VCS/version management tool is Git.					
	Accept any suitable example.					

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
9	**Award 1 mark for each correct response to a maximum of 6 marks. Do not accept the reverse points more than once.**					8
	Different sorting algorithms have varying time and space complexities.	1				
	Depending on the size and type of the data to be sorted different sorting algorithms need to be chosen.	1				
	Recursive sorting algorithms operate by dividing the dataset into smaller sub-datasets and then recursively sorting those datasets.	1				
	Recursive sorting algorithms have good time complexity for larger datasets.	1				
	Recursive sorting algorithms are often easier to implement.	1				
	Recursive can have higher space requirements given they make use of recursive calls.	1	1b			
	Non-recursive sorting algorithms operate by iteratively rearranging the dataset until all elements are sorted.	1				
	Non-recursive sorting algorithms are usually faster than recursive algorithms for smaller datasets.	1				
	Non-recursive sorting algorithms usually have lower space requirements.	1				
	Award 1 mark for each example to a maximum of two					
	An example of a recursive algorithms is a Merge Sort or Quicksort.	1				
	An example of a non-recursive algorithms is an Insertion Sort or Bubble Sort.	1				
	Accept any suitable example(s).					

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
10	00001111_2 $\underline{10000000_2}$ AND **0**0000000 State of sign bit is **0** which is **positive** Award 1 mark for using an AND or XOR mask. Award 1 mark for correct result Award 1 mark for determining the state of sign bit.	 1 1 1		2a		3
11 (a)	1 mark for identifying i loop will execute n times i.e. n 1 mark for identifying j loop will execute n times i.e. 2n 1 mark for identifying 3 instructions executed 3 marks for correctly identifying 2n + 3 1 mark for determining that the order will be dominated by n 1 mark for determining that the growth rate for time performance in Big O notation is O(n)	1 1 1 3 1 1			3c	5
11 (b)	• The algorithm uses three data structures, three one-dimensional arrays. • Two input arrays of size n are combined to create a third. • Therefore, total storage requirements = n. Given this, the growth rate for memory will be O(n).	1 1		2b		2
11 (c)	**Linear Complexity O(n)** Time and Size axes labelled correctly Correct gradient of line	 1 1		2b		2

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
12	**Award 1 mark for each correct response up to a maximum of 3 marks.** • A shortest path algorithm is a method used to find the shortest path between two nodes or points. • It achieves this by calculating the distance/weight between nodes and selecting the path with the lowest total distance/cost. • A shortest path algorithm calculates a value for the distance from the starting node to each adjacent node. • The distance value is then updated if a shorter path is found. • This process is repeated until the destination node is reached. • A shortest path algorithm can be used in network routing / transportation (GPS) to find the most efficient path. **Award 1 mark for correct example** An example of a shortest path algorithm is Dijkstra's algorithm. **Accept any suitable example.**	1 1 1 1 1 1 1	1b			4
13	**Indicative content** Natural language interfaces enable communication with computers in a more intuitive and user-friendly manner. They can facilitate communication for individuals with disabilities, such as those with motor or visual impairments. Natural language interfaces can reduce the need for training and technical expertise, making technology more accessible to the public. These interfaces simplify complex tasks, such as data analysis and software programming, by allowing users to express their intentions in natural language. These interfaces have the potential to improve the efficiency of user interactions with computers by reducing the need for input.		1b			12

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
	They can enhance the personalisation of computer systems by allowing users to express preferences and constraints in natural language.					
	Natural language interfaces can enable more efficient communication in industries such as healthcare, where time and accuracy are crucial.					
	They can improve the accuracy of information retrieval by allowing users to express their queries in natural language, rather than relying on pre-defined keywords.					
	These interfaces can enable more natural and engaging communication in virtual and augmented reality applications.					
	Natural language interfaces can provide a more seamless integration of voice and text-based communication channels such as in smart speakers and phones.					
	As natural language processing and AI technology advances, they will become more capable of understanding the nuances and ambiguities in human language.					
	Words can have multiple meanings. For example, "bat" can refer to a flying mammal or a piece of sports equipment.					
	Phrases can be interpreted in different ways based on context. For example, "bank" can mean a financial institution or the side of a river.					
	Different people may express the same idea in different ways, using synonyms or completely different phrases.					
	Idioms and Colloquialisms					
	Detecting sarcasm and irony is challenging because it often relies on tone of voice or context clues.					
	Accents and Dialects, variations in pronunciation can affect the accuracy of speech recognition systems.					
	Homophones, words that sound the same but have different meanings, like "there", "their", and "they're", can cause confusion.					

Question	Answer	Mark	AO1	AO2	AO3	TOTAL
	Understanding cultural references or jokes requires knowledge of the relevant culture.					
	New words and slang are constantly being created, which can be challenging for computers to keep up with.					
	Machine Learning: Using large datasets to train models that can better understand and predict human language patterns.					
	Contextual Models: Implementing models like GPT-4 that consider context to interpret language more accurately.					
	Continuous Learning: Systems that can learn and adapt over time, improving their understanding of new words and usage patterns.					
	Multimodal Inputs: Combining text with other inputs like voice tone, facial expressions, and body language to better understand intent and emotion.					
	Total					**100**

Band	Q12 AO1b - Max 12 marks
3	**9 – 12 marks** **The candidate has:** • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the topics as specified in the indicative content. Clear knowledge is defined as responses that provide relevant detailed points of the indicative content, which relate to an extensive amount of the indicative content. • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • effectively drawn together different areas of knowledge, skills and understanding from all relevant areas across the course of study • used appropriate technical terminology confidently and accurately.
2	**5 - 8 marks** **The candidate has:** • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topics as specified in the indicative content. Satisfactory knowledge is defined as responses that provide relevant points of the indicative content, which relate to the indicative content. • presented a discussion with limited examples • drawn together different areas of knowledge, skills and understanding from a number of areas across the course of study • used appropriate technical terminology.
1	**1- 4 marks** **The candidate has:** • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as responses that provide limited relevant points of the indicative content, which relate to a limited amount the indicative content. • used limited technical terminology.
0	Response not credit worthy or not attempted.