GCE A LEVEL MARKING SCHEME

AUTUMN 2020

A LEVEL
COMPUTER SCIENCE - COMPONENT 1
A500U10-1

INTRODUCTION

This marking scheme was used by WJEC for the 2020 examination. It was finalised after detailed discussion at examiners' conferences by all the examiners involved in the assessment. The conference was held shortly after the paper was taken so that reference could be made to the full range of candidates' responses, with photocopied scripts forming the basis of discussion. The aim of the conference was to ensure that the marking scheme was interpreted and applied in the same way by all examiners.

It is hoped that this information will be of assistance to centres but it is recognised at the same time that, without the benefit of participation in the examiners' conference, teachers may have different views on certain matters of detail or interpretation.

WJEC regrets that it cannot enter into any discussion or correspondence about this marking scheme.
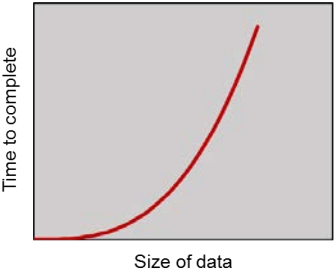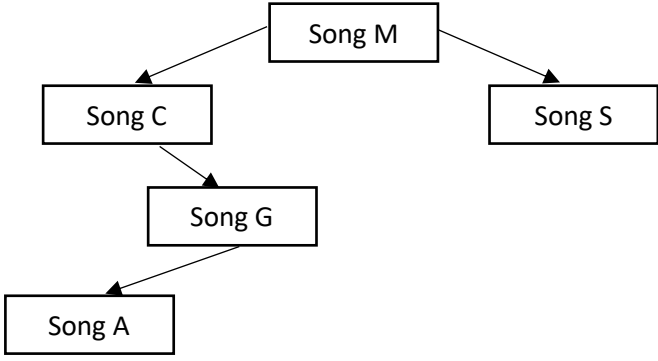
GCE A LEVEL COMPUTER SCIENCE

AUTUMN 2020 MARK SCHEME

Question	Answer	Mark	AO1	AO2	AO3	Total
1. (a)	One mark for each point up to a maximum of 4.					4
	A recursive algorithm calls itself (1 mark) using a parameter (1 mark) and has a stopping condition (1 mark).	3	1.1			
	Any suitable example of a recursive algorithm e.g. quicksort.	1	1.1			
(b)	Non-recursive algorithms are useful when a data structure is fixed in size like an array.	1	1.1			2
	Non-recursion can reduce time complexity in sorting algorithms if implemented efficiently.	1	1.1			
	Non-recursive solutions require less memory than recursive solutions reducing the demand on resources.	1	1.1			
	Non-recursive solutions are easier to write.	1	1.1			

Question	Answer	Mark	AO1	AO2	AO3	Total
2. (a)	Correct answer can be established using different steps / laws / rules / identities / dual relations. One possible solution: $\overline{Z} + (\overline{Z + X}) + X + Y + \overline{Z.Y}$ $\overline{Z} + (\overline{Z} . \overline{X}) + X + Y + \overline{Z.Y}$ $\overline{Z} + (\overline{Z} . \overline{X}) + X + Y + \overline{Z} + \overline{Y}$ $\overline{Z} + X + Y + \overline{Z} + \overline{Y}$ $\overline{Z} + X + 1$ 1					5
	Correctly applying De Morgan's Law 1 mark	1		2.1		
	Correctly applying identities to arrive at correct answer 4 marks	4		2.1		
	Correctly applying identities but arriving at wrong answer 1 mark					
(b)	Correct answer can be established using different steps / laws / rules / identities / dual relations. One possible solution: $A . (A + C) . B . (B + C) + A$ $A . B . (B + C) + A$ $A . B . B + A$ $A + A . B . B$ $A + A . B$ A					5
	Correctly applying identities to arrive at correct answer 5 marks Correctly applying identities but arriving at wrong answer 1 mark	5		2.1		

Question	Answer	Mark	AO1	AO2	AO3	Total
3.	Award one mark for each correct description and one mark for each appropriate example. Max of 8 marks.					8
	• Self-documenting identifiers allows code to be followed and understood more easily.	1				
	• Self-documenting identifiers reduces the need for additional documentation to be produces such as additional annotation or software manuals.	1				
	• An example of a self-document identifier would be using an appropriate variable name such as:	1	1.1			

	o int VAT = 20'.
	   o float f = 23.0
	   o String s = "Hello World"
		1				

	• Program layout allows blocks of code and constructs to be followed and identified more easily.	1				
	• A consistent program layout helps improve the quality of the software and allows developers to maintain quality and standards.	1				
	• An example of program layout could be correctly using indentation to identify the start and end of constructs such as:	1	1.1			

	o IF statements
	   o Loop / nested loop structures
	   o String s = "Hello World"
		1				

	• Annotation is important as it allows developers to record the development process and logic with the actual code.	1				
	• This is important as many developers could be working on the one project and each developer needs to understand the logic between one and others code.	1				
	• An example of annotation would be to demonstrate logic such as:	1	1.1			

	o 'X DIV 2 //calculate if X is even/odd'
	   o # Use of print() here
	   o /* Person Class @return String name */
		1				

Question	Answer	Mark	AO1	AO2	AO3	Total
4. (a)	<table><tr><td>A</td><td>B</td><td>A AND B</td><td>A OR (A AND B)</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> Award one mark for each correct column	2		2.1		2
(b)	<table><tr><td>A</td><td>B</td><td>C</td><td>(A + B)</td><td>(A + C)</td><td>(A + B).(A + C)</td><td>(B.C)</td><td>A + (B.C)</td></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table> Award one mark for correct combinations of A, B and C Award one mark for each correct column	1 5		2.1		6

Question	Answer	Mark	AO1	AO2	AO3	Total
5. (a)	1 mark for identifying i loop will execute n / n – 1 times	1			3.1	5
	1 mark for identifying j loop will execute n^2 times	1			3.1	
	1 mark for correct numbers of operations n^2 + 5	1			3.1	
	1 mark for determining that the order will be dominated by n^2	1			3.1	
	1 mark for determining that the growth rate for time performance is $O(n^2)$	1			3.1	
(b)	The algorithm only uses one data structure, a two-dimensional array. Therefore, total storage requirements = 1.	1			3.1	2
	As only one data structure is being used, the growth rate for memory will be constant O(1).	1			3.1	
(c)	Identifying polynomial complexity Time axis labelled correctly Size axis labelled correctly Correct gradient of line	1 1 1 1		2.1 2.1 2.1 2.1		4
6.	<day> ::= 01|02|03 . . . 30|31 <month> ::= JAN|FEB|MAR . . . NOV|DEC <year> ::= 2020|2021|2022 . . . 9998|9999 <digit> ::= 0|1|2 . . . 8|9	1		2.1		5
	<longitudinal> ::= -180|-179|-178 . . . 179|180 <latitudinal> ::= -90|-89|-88 . . . 89|90	1				
	<digits>::= <digit><digit><digit> <digit><digit><digit> <filename>::= <day><month> <year>_< longitudinal >_ <digits>_ <latitudinal>_ <digits>.PNG	1 1 1		2.1		
	Answer not correct if BNF notation used incorrectly. Must include underscore (_) for full marks.					

Polynomial Complexity
$O(n^2)$
Time to complete
Size of data

Question	Answer	Mark	AO1	AO2	AO3	Total
7. (a)						2
	1 mark for suitable ordered examples and pointers 1 mark for unbalanced tree	1 1		2.1 2.1		
(b)	Any of the following up to a maximum of 3.					3
	The most suitable way to traverse the tree is inorder.	1		2.1		
	Inorder traversal starts with the left subtree nodes being visited first.	1		2.1		
	Then visit the root node and finally the right subtree nodes.	1		2.1		
	Inorder allows every node to be visited in sorted order.	1		2.1		
(c)	**Indicative content**					4
	A queue would be the most suitable data structure to store each playlist.	1		2.1		
	A queue follows the first in first out (FIFO/LILO) principle.	1		2.1		
	Data is added (enqueuing) at the rear end of the structure.	1		2.1		
	Data is accessed and removed (dequeuing) from the front of the structure which is suitable for storing a sequential playlist. Accept any suitably justified data structure.	1		2.1		

Question	Answer	Mark	AO1	AO2	AO3	Total
8.	**Indicative content**					8
	Waterfall approach					
	Sequential process	1	1.1			
	Developers draft the design of a system up front and it does not change.	1	1.1			
	Once the analysis and design stages are complete, developers cannot go back to a to make any changes.	1	1.1			
	If the analysis or design of the project are inaccurate or incorrect in any way, the project will fail due to the rigidness of the waterfall methodology.	1	1.1			
	Requires less communication between the client and the developer.	1	1.1			
	Client input is only required during analysis and at times the design stage.	1	1.1			
	The Agile approach					
	Incremental approach to development					
	Developers start with a simple project design and requirements.	1	1.1			
	Iterative approach as analysis and design relies on each other.	1	1.1			
	Analysis informs design and the design informs further analysis to be undertaken.	1	1.1			
	Changes can be made after each phase of development, analysis can be revisited, and designs changed.	1	1.1			
	Strong communication between the client and the developer should be regular	1	1.1			
	Clients are involved during all stages of development.	1	1.1			

Question	Answer	Mark	AO1	AO2	AO3	Total
9. (a)	Max mark of 4 from the following (max of two from each):		1.1			4
	Advantages					
	• Improved productivity when developing software due to the flexible and extendable nature of OOP.	1				
	• Software is easier to maintain as OOP is modular and reusable.	1				
	• Development is faster due the reusable code and libraries.	1				
	• Development is cheaper.	1				
	• Software can be tested more easily making it more high quality.	1				
	• Software is easier to design as model the real world.	1				
	Disadvantages					
	• OOP is difficult and not as 'logical' to some developers, it is complex to create application in.	1				
	• Software can become larger – more code - than procedural programs.	1				
	• OOP programs can run slower than PP as there is more code to execute.	1				
	• OOP cannot be used for all types of software application such as machine learning and AI.	1				
	• OOP can be difficult to debug.	1				
(b)	A class is a template or blueprint for a specific object. It defines an object's instance variables (attributes/properties) and behaviour (methods). An object is an instance of a class.					4
	1 mark for template/blueprint/contract	1	1.1			
	1 mark for defines variables (attributes/properties)	1	1.1			
	1 mark for defines behaviour (methods)	1	1.1			
	1 mark for stating an object is an instance of a class	1	1.1			

Question	Answer	Mark	AO1	AO2	AO3	Total
(c)	A method is a programmed behaviour/subroutine that is included in an object of a class. A method can only access data within its own object (encapsulation).					2
	1 mark for a method is a programmed behaviour/subroutine	1	1.1			
	1 mark for stating a method can only access its own objects data (encapsulation).	1	1.1			

Question	Answer	Mark	AO1	AO2	AO3	Total
10.	**Indicative content**					9

```
1   Declare MyArray[0 to 6]
2   Declare Start is integer
3   Declare End is integer
4   Declare Found is Boolean
5   Declare Mid is integer
6
7   set Start = 0
8   set End = 6
9   set Found = False
10
11  input SearchValue
12
13  repeat
14      set Mid = (Start + End) DIV 2
15      if SearchValue = MyArray[Mid]
        then
16              set Found = True
17              Output "SearchValue found at
                position", Mid
18      endif
19
20      if SearchValue > MyArray[Mid]
        then
21              set Start = Mid + 1
22      endif
23
24      if SearchValue < MyArray[Mid]
        then
25              set End = Mid - 1
26      endif
27  until (Found = True) OR (End < Start)
28
29      if Found = False
30          Output "SearchValue not
    found"
31      endif
```

Question	Answer	Mark	AO1	AO2	AO3	Total
	Declare and initialise variables	1			3.3	
	Input SearchValue	1			3.3	
	Loop structure and increment	1			3.3	
	Comparison with searchValue and outpout position if found	1			3.3	
	Correct terminating condition for loop	1			3.3	
	Correctly discard half of array if myArray(Mid)>SearchValue	1			3.3	
	Correctly discard half of array if myArray(Mid)<SearchValue	1			3.3	
	Output message of found	1			3.3	
	Fully functional algorithm	1			3.3	

Question	Answer	Mark	AO1	AO2	AO3	Total
11.	**Indicative content** 	Input	Output
---	---
a	error message 1
exam@@eduqas	error message 2 & 3
exam@eduqascom	error message 3
exam@eduqas.com	valid | 1 mark for each correct example	4		2.1		4
12.	**Indicative content** • The factors that need to be considered when proposing a new system solution include cost, time scale and budget. • A proposed system should be cost effective, in terms of human resources, finances, technology and time. • A proposed system needs to be effective in terms of human resource costs. The proposed system must not over or under utilise developers. When developing a new system, each developer should be allocated roles and development activities. • These activities should be overseen by a lead developer to ensure that human resources are being fully utilised and cost effective. • A proposed project should be financially cost effective. Developers should research and source the most financially cost-effective methods/resources/technologies when proposing a new system. • Technologies sources including hardware and software should be cost effective. • The system should have a specific time scale for development from inception to evaluation. • The proposed system should follow a suitable development methodology with appropriate and realistic deadlines. • These deadlines should follow a suitable plan to ensure an effective time scale for the project. • The system needs to have a controlled budget. This budget should be managed accordingly to ensure the success and economic viability of the project. • When implementing a new solution there are various methods of changeover that can be employed including direct, pilot, phased and parallel.		1.1			12

Question	Answer	Mark	AO1	AO2	AO3	Total
	• Direct changeover is the simplest but most risky method of changeover. This method should only be employed where there is not an existing system already in place. • New systems always come with a variety of problems including bug and compatibility issues and directly changing to a new system could have a significant impact on business and productivity if these issues occur. • Pilot changeover is usually employed when a business has the required amount of resources to effectively test a new system by deploying it into one area for example, a new stock management system in one of a company's many warehouses. • This method allows bugs and other issues to be confined to just one area and when fixed the system can be rolled out on a much larger scale. • Phased changeover is used when a system can be deployed in units or modules. This works well when parts of a new system are being developed independently and upgrading an existing system. • When each module is implemented into an existing system many compatibility issues can occur between the new systems modules and the existing system. • Parallel changeover is used when there can be opportunity for a system to fail. Phased changeover implements a new system alongside an existing system and if one fails the other takes over. • System tasks are run concurrently on both the new system and the existing causing a duplication of tasks. These tasks can be used to ensure consistency between the new and existing systems. • Parallel changeover is employed for critical systems such as those in hospitals and banks where data access and integrity is critical.					

Band	Q13 AO1b - Max 12 marks
3	**10-12 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 on the costs of proposing a new solution and methods of changeover, 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	**5-9 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 on the costs of proposing a new solution and methods of changeover, 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 on the costs of proposing a new solution and methods of changeover, which relate to a limited amount the indicative content. • used limited technical terminology.
0	Response not credit worthy or not attempted.