GCE

Computer Science

H446/02: Algorithms and programming

Advanced GCE

Mark Scheme for June 2019

OCR (Oxford Cambridge and RSA) is a leading UK awarding body, providing a wide range of qualifications to meet the needs of candidates of all ages and abilities. OCR qualifications include AS/A Levels, Diplomas, GCSEs, Cambridge Nationals, Cambridge Technicals, Functional Skills, Key Skills, Entry Level qualifications, NVQs and vocational qualifications in areas such as IT, business, languages, teaching/training, administration and secretarial skills.

It is also responsible for developing new specifications to meet national requirements and the needs of students and teachers. OCR is a not-for-profit organisation; any surplus made is invested back into the establishment to help towards the development of qualifications and support, which keep pace with the changing needs of today's society.

This mark scheme is published as an aid to teachers and students, to indicate the requirements of the examination. It shows the basis on which marks were awarded by examiners. It does not indicate the details of the discussions which took place at an examiners' meeting before marking commenced.

All examiners are instructed that alternative correct answers and unexpected approaches in candidates' scripts must be given marks that fairly reflect the relevant knowledge and skills demonstrated.

Mark schemes should be read in conjunction with the published question papers and the report on the examination.

© OCR 2019

Annotations

Annotation	Meaning
	Omission mark
	Benefit of the doubt
	Incorrect point
	Follow through
	Not answered question
	No benefit of doubt given
	Repeat
	Correct point
	Too vague
	Zero (big)
	Level 1
	Level 2
	Level 3

Subject-specific Marking Instructions

INTRODUCTION

Your first task as an Examiner is to become thoroughly familiar with the material on which the examination depends. This material includes:

- the specification, especially the assessment objectives
- the question paper and its rubrics
- the mark scheme.

You should ensure that you have copies of these materials.

You should ensure also that you are familiar with the administrative procedures related to the marking process. These are set out in the OCR booklet **Instructions for Examiners**. If you are examining for the first time, please read carefully **Appendix 5 Introduction to Script Marking: Notes for New Examiners**.

Please ask for help or guidance whenever you need it. Your first point of contact is your Team Leader.

USING THE MARK SCHEME

Please study this Mark Scheme carefully. The Mark Scheme is an integral part of the process that begins with the setting of the question paper and ends with the awarding of grades. Question papers and Mark Schemes are developed in association with each other so that issues of differentiation and positive achievement can be addressed from the very start.

This Mark Scheme is a working document; it is not exhaustive; it does not provide 'correct' answers. The Mark Scheme can only provide 'best guesses' about how the question will work out, and it is subject to revision after we have looked at a wide range of scripts.

The Examiners' Standardisation Meeting will ensure that the Mark Scheme covers the range of candidates' responses to the questions, and that all Examiners understand and apply the Mark Scheme in the same way. The Mark Scheme will be discussed and amended at the meeting, and administrative procedures will be confirmed. Co-ordination scripts will be issued at the meeting to exemplify aspects of candidates' responses and achievements; the co-ordination scripts then become part of this Mark Scheme.

Before the Standardisation Meeting, you should read and mark in pencil a number of scripts, in order to gain an impression of the range of responses and achievement that may be expected.

In your marking, you will encounter valid responses which are not covered by the Mark Scheme: these responses must be credited. You will encounter answers which fall outside the 'target range' of Bands for the paper which you are marking. Please mark these answers according to the marking criteria.

Please read carefully all the scripts in your allocation and make every effort to look positively for achievement throughout the ability range. Always be prepared to use the full range of marks.

LEVELS OF RESPONSE QUESTIONS:

The indicative content indicates the expected parameters for candidates' answers, but be prepared to recognise and credit unexpected approaches where they show relevance.
Using 'best-fit', decide first which set of BAND DESCRIPTORS best describes the overall quality of the answer. Once the band is located, adjust the mark concentrating on features of the answer which make it stronger or weaker following the guidelines for refinement.

- **Highest mark:** If clear evidence of all the qualities in the band descriptors is shown, the HIGHEST Mark should be awarded.
- **Lowest mark:** If the answer shows the candidate to be borderline (i.e. they have achieved all the qualities of the bands below and show limited evidence of meeting the criteria of the band in question) the LOWEST mark should be awarded.
- **Middle mark:** This mark should be used for candidates who are secure in the band. They are not 'borderline' but they have only achieved some of the qualities in the band descriptors.

Be prepared to use the full range of marks. Do not reserve (e.g.) high Band 3 marks 'in case' something turns up of a quality you have not yet seen. If an answer gives clear evidence of the qualities described in the band descriptors, reward appropriately.

	AO1	AO2	AO3
High (thorough)	Precision in the use of question terminology. Knowledge shown is consistent and well-developed. Clear appreciation of the question from a range of different perspectives making extensive use of acquired knowledge and understanding.	Knowledge and understanding shown is consistently applied to context enabling a logical and sustained argument to develop. Examples used enhance rather than detract from response.	Concerted effort is made to consider all aspects of a system / problem or weigh up both sides to an argument before forming an overall conclusion. Judgements made are based on appropriate and concise arguments that have been developed in response resulting in them being both supported and realistic.
Middle (reasonable)	Awareness of the meaning of the terms in the question. Knowledge is sound and effectively demonstrated. Demands of question understood although at times opportunities to make use of acquired knowledge and understanding not always taken.	Knowledge and understanding applied to context. Whilst clear evidence that an argument builds and develops through response there are times when opportunities are missed to use an example or relate an aspect of knowledge or understanding to the context provided.	There is a reasonable attempt to reach a conclusion considering aspects of a system / problem or weighing up both sides of an argument. However the impact of the conclusion is often lessened by a lack of supported judgements which accompany it. This inability to build on and develop lines of argument as developed in the response can detract from the overall quality of the response.
Low (basic)	Confusion and inability to deconstruct terminology as used in the question. Knowledge partial and superficial. Focus on question narrow and often one-dimensional.	Inability to apply knowledge and understanding in any sustained way to context resulting in tenuous and unsupported statements being made. Examples if used are for the most part irrelevant and unsubstantiated.	Little or no attempt to prioritise or weigh up factors during course of answer. Conclusion is often dislocated from response and any judgements lack substance due in part to the basic level of argument that has been demonstrated throughout response.

	Assessment Objective
AO1	Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.
AO1.1	Demonstrate **knowledge** of the principles and concepts of abstraction, logic, algorithms, data representation or other as appropriate.
AO1.2	Demonstrate **understanding** of the principles and concepts of abstraction, logic, algorithms, data representation or other as appropriate.
AO2	Apply knowledge and understanding of the principles and concepts of computer science including to analyse problems in computational terms.
AO2.1	Apply knowledge and understanding of the principles and concepts of computer science.
AO2.2	Analyse problems in computational terms.
AO3	Design, program and evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.
AO3.1	Design computer systems that solve problems.
AO3.2	Program computer systems that solve problems.
AO3.3	Evaluate computer systems that solve problems, making reasoned judgements about these and presenting conclusions.

Question	Answer	Marks	Guidance
1a	1 mark per bullet • Queue outputs data in a First In First Out fashion • It will retrieve the temperature values in the order they were recorded or • Stack outputs the data in a Last In First Out fashion • It will retrieve the temperature values in the reverse of the order they were recorded	**2** **AO1.2 (1)** **AO2.2 (1)**	Mark Point 1 is the definition Mark Point 2 is for context of the temperature values
1bi	It returns a value	**1** **AO2.1 (1)**	

1bii	1 mark per completed word
	
	`processedData[0] = 0`
	
	`firstDay = dequeue()`
	
	`for count = 1 to 6`
	
	`    processedData[count] = dequeue() - firstDay`
	
	`next count`
		3
		
		AO2.2 (1)
		AO3.2 (2)
			Exact answers only

<table>
<tr>
<td>1biii</td>
<td>1 mark per bullet to max 5.
Max 3 if no application to data in <code>processedData</code>

Compares each pair of data e.g. 0 and 0.5
If they are in the correct order it moves to the next pair e.g. 0.5 and 0
If they are in the wrong order it swaps them e.g. 0.5 and 0 becomes 0 and 0.5
Continues to the end of the array e.g. Pass 1 complete
If there has been a swap it checks again e.g. Pass 2 complete
If there have been no swaps it is sorted

<table>
<tr><td>0</td><td>0.5</td><td>0</td><td>1</td><td>2</td><td>1.5</td><td>1</td><td rowspan="4">Pass 1</td></tr>
<tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>2</td><td>1.5</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>2</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1.5</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1</td><td>1.5</td><td>2</td><td>Pass 2</td></tr>
<tr><td>0</td><td>0</td><td>0.5</td><td>1</td><td>1</td><td>1.5</td><td>2</td><td>No swaps</td></tr>
</table>
</td>
<td>5

AO1.1 (2)
AO1.2 (1)
AO2.1 (1)
AO2.2 (1)</td>
<td>Allow (full) credit for tables showing the bubblesort being completed.</td>
</tr>
<tr>
<td>1biv</td>
<td>1 mark per bullet
O(n)

<u>Linear</u>
Best time grows at the same rate as the number of elements
This is the case when the data is already in order

O(n^2)

<u>Polynomial</u> / <u>Quadratic</u>
Worst and average time is proportional to the square (polynomial) of the number of elements
Worst case is when the data is initially in the reverse order

O(1)

<u>Constant</u>
Will always take the same amount of memory (in addition to the list itself).

</td>
<td>6

AO1.1 (3)
AO1.2 (3)</td>
<td>Note: First Mark Point is for the identification, second Mark Point is for the description

Note: Do not allow descriptions relating to time complexity for 'Worst Space O(1)'

Note: Do not allow ‘equal to’ in descriptions, O(n) and O(n^2) grow in proportion to the number of items</td>
</tr>
</table>

2ai	Binary Tree / Binary Search Tree	1 **AO2.1 (1)**	
2aii	1 mark per bullet • 1st layer: Lily • 2nd layer: Daisy, Sunflower • 3rd layer: Begonia, Hosta, Peony 4th layer: Rose	3 **AO1.2 (1)** **AO2.1 (1)** **AO2.2 (1)**	
2aiii	1 mark per bullet to max 4. Max 2 marks for no application to the tree. • Depth first starts at the root (Lily) • …and goes all the way down one branch to the bottom (Begonia) • It stores which nodes it has visited / pushes nodes visited onto a stack • When it cannot go any further • …It then backtracks/returns to the previous node • And continues to backtrack until a node is reached with unvisited children. • …and checks down that branch • In the tree shown, after visiting Begonia, the algorithm would backtrack to Daisy… • …and would then visit Hosta *(Accept any other example)*	4 **AO1.1 (1)** **AO1.2 (1)** **AO2.1 (1)** **AO2.2 (1)**	

<table>
<tr>
<td>2bi</td>
<td>1 mark per bullet
• Correct NextPointer values
• Suitable end/null pointer

<table>
<tr><th>Data item</th><th>Data</th><th>NextPointer</th></tr>
<tr><td>0</td><td>Begonia</td><td>1</td></tr>
<tr><td>1</td><td>Daisy</td><td>2</td></tr>
<tr><td>2</td><td>Hosta</td><td>3</td></tr>
<tr><td>3</td><td>Lily</td><td>4</td></tr>
<tr><td>4</td><td>Peony</td><td>5</td></tr>
<tr><td>5</td><td>Rose</td><td>6</td></tr>
<tr><td>6</td><td>Sunflower</td><td>null</td></tr>
</table>
</td>
<td>2

AO2.1 (2)</td>
<td>Exact values only. Allow -1 for null pointer or equivalent such as Φ.
Do not allow a blank or 0.</td>
</tr>
<tr>
<td>2bii</td>
<td>1 mark per bullet
• Lavender added in position 7
• …Hosta points to 7
• …Lavender points to 3

<table>
<tr><th>Data item</th><th>Data</th><th>NextPointer</th></tr>
<tr><td>0</td><td>Begonia</td><td>1</td></tr>
<tr><td>1</td><td>Daisy</td><td>2</td></tr>
<tr><td>2</td><td>Hosta</td><td>7</td></tr>
<tr><td>3</td><td>Lily</td><td>4</td></tr>
<tr><td>4</td><td>Peony</td><td>5</td></tr>
<tr><td>5</td><td>Rose</td><td>6</td></tr>
<tr><td>6</td><td>Sunflower</td><td>null</td></tr>
<tr><td>7</td><td>Lavender</td><td>3</td></tr>
</table>
</td>
<td>3

AO1.2 (1)
AO2.2 (2)</td>
<td>Do not credit answers that do not place lavender in position 7 and then update pointer positions</td>
</tr>
</table>

2biii	1 mark per bullet • Traverse the list to the item immediately prior to the item to be removed (1) • … which is DataItem 1 - Daisy • Find the value of the NextPointer of the item to be removed • … which is the NextPointer of DataItem 2 - Hosta, value 7 • Set the nextPointer of the item prior to the item to be removed to the NextPointer value of the DataItem to be removed • … update the NextPointer of DataItem 1 - Daisy from 2 to 7 (Lavender)	4 **AO1.1 (1)** **AO1.2 (1)** **AO2.2 (2)**	Find the *item before* item to be deleted (Daisy) Find nextPtr of item to be deleted (Hosta) Update nextPtr of the *item before (Daisy)* to the nextPtr of item to be deleted (Hosta) i.e. Daisy 2 is updated to Daisy 7 Allow FT from 2b(ii/iii) if candidate has used table in fig 2.1 (e.g. Daisy would now point to Lily at position 3)

2biv	1 mark per bullet
	• Start at the `firstElement` in the list
	• Correctly looping until null pointer found / end of list
	• Outputting the data element
	• Accessing the pointer to the next element
	• Appropriate comment(s)
	
	e.g.
	`currentElement = firstElement`
	`while(currentElement != null) //Continue until last node`
	`  print(plantList[currentElement,0])`
	`  currentElement = plantList[currentElement,1]`
	`endwhile`
		5
		
		AO2.1 (1)
		AO2.2 (2)
		AO3.2 (2)
			Note: Solution must utilise pointers in a linked list; it cannot use a FOR loop as the number of elements is not known and the data is not in order by index number
			
			Note: identifiers given in the question as `plantList` and `firstElement` should be used accurately in the solution
			
			Note: allow credit for answers that interpret the data structure as an array of records/structures with data/pointer fields

3ai	<u>if</u>	1 **AO1.1 (1)**	
3aii	1 mark per bullet • Branching decides which code is run / only runs code once • Iteration repeatedly runs the same code in the same sequence	2 **AO1.2 (2)**	
3aiii	num1, num2	1 **AO2.1 (1)**	Exact identifier names required
3aiv	1 mark per bullet • By Value • … the original values do not need to be modified • … byRef would not work / would cause the routine to crash	2 **AO2.2 (2)**	
3av	1 mark per bullet • Gives the remainder after division • E.g. 10 MOD 3 = 1	2 **AO1.1 (1)** **AO1.2 (1)**	
3b	1 mark per bullet to max 3 • Num2 != 0 therefore return GCD(20,10) • Num2 != 0 therefore return GCD(10,0) • Final return value = 10	3 **AO2.1 (1)** **AO2.2 (2)**	Allow FT for numerical errors

3ci	1 mark for benefit, 1 mark for drawback Benefit: • The program can/might run faster • Cannot run out of stack space/memory • Easier to trace/follow Drawback: • Iteration can lead to lengthier code • Iteration can lead to code that looks more complex / is harder to understand • some problems are more elegantly coded with a recursive solution	2 **AO1.1 (1)**	

3cii	1 mark for each correct statement
	
	`function newGCD(num1, num2)`
	`  temp = 0`
	`  while (num2 != 0)`
	`  temp = num2`
	`  num2 = num1 MOD num2`
	`  num1 = temp`
	`  endwhile`
	`  return num1`
	`endfunction`
		4
		
		AO2.2 (2)
		AO3.2 (2)	

4a	1 mark per bullet • She can split the problem down into sub problems • It will creates a more manageable problem / simpler to understand / maintain • can tackle each sub problem independently	2 **AO1.2 (1)** **AO2.2 (1)**	
4bi	1 mark per bullet, max 2 per sub-procedure e.g. • Select character (name, gender) • Gives the user options for choosing a character • Choose level • Give the user the choice of level (easy, normal, challenging) and take the user input • Touch enemy • Called to determine if the character touches an enemy • Lose life • Remove a life, if <0 then game over • End level • Move onto next level One mark for identifying sensible subroutine, 1 mark for description	6 **AO2.1 (2)** **AO2.2 (2)** **AO3.2 (2)**	Do not award any user *input* related procedures e.g. Left/Right input (but character movement *output* on screen left/right would be valid) Allow other reasonable responses from the scenario e.g. generate enemy()
4bii	1 mark per bullet • Decision based on what the user has input • E.g. If they click left move the character left // if they click right move the character right // if they click space bar make the character jump	2 **AO2.1 (1)** **AO2.2 (1)**	

<table>
<tr>
<td>4biii</td>
<td>1 mark per bullet
• The result from one process / procedure feeds into the next
• E.g. the result of detecting a character touching an enemy feeds into reducing the number of lives</td>
<td>2

AO1.2 (1)
AO2.2 (1)</td>
<td>Note: 1 Mark Max for a generic description of pipelining</td>
</tr>
<tr>
<td>4c</td>
<td>1 mark for final solution, max 5 for showing the stages

Mark A as the current node initially
Record B = 1, C = 2 (mark A as visited)
Record E = 5 (and mark B as visited)
(Record D = 3, F = 5 (and mark B as visited)
Change E to 4 (overriding previous value, and mark D as visited)
Record G = 6 (and mark E as visited)
…Do not change G as greater than current (mark F as visited)
(G as visited) H = 10 (Mark G as visited)
Solution: A-C-D-E-G-H path length 10

<table>
<tr><td>Node</td><td>Visited</td><td>From A</td><td>Previous Node</td><td></td></tr>
<tr><td>A</td><td>✓</td><td>0</td><td>-</td><td>1 Mark</td></tr>
<tr><td>B</td><td>✓</td><td>1</td><td>A</td><td rowspan="2">1 Mark</td></tr>
<tr><td>C</td><td>✓</td><td>2</td><td>A</td></tr>
<tr><td>D</td><td>✓</td><td>3</td><td>C</td><td rowspan="2">1 Mark</td></tr>
<tr><td>F</td><td>✓</td><td>5</td><td>C</td></tr>
<tr><td>E</td><td>✓</td><td><s>5</s> 4</td><td><s>B</s> D</td><td>2 Marks
Initial visit, plus override values</td></tr>
<tr><td>G</td><td>✓</td><td>6</td><td>E</td><td>1 Mark</td></tr>
<tr><td>H</td><td></td><td>10</td><td>G</td><td>1 Mark</td></tr>
</table></td>
<td>6

AO1.2 (1)
AO2.2 (3)
AO2.2 (2)</td>
<td>Guidance – 1 mark only for stating the solution of A-C-D-E-G-H length 10</td>
</tr>
</table>

4d	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of IDEs; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate is able to weigh up the context which results in a supported and realistic judgment as to whether IDEs are useful in this context. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of IDEs; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether IDEs are useful in this context. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of IDEs with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** Tools to aid writing • Coloured font • Predictive text • Auto-correct Tools to aid de-bugging • Stepping • Break points • Variable watch window **AO2: Application** e.g. • Can write subroutines for the program and it will tell you what parameters are needed • Allow you to run the program without exiting the software / having to load a separate compiler • Integrates other tools such as version control. • Can reduce spelling errors • Can use to fix errors that might occur / debug **AO3: Evaluation** e.g. • User friendly for novices • Increase speed of writing • Fewer mistakes • Increase speed of testing / finding errors • Collaborative team working facilitated

5a	1 mark per bullet to max 6. Max 4 if generic description given with no application Max 4 if a diagrammatic solution is given with no description • Splits the list in half repeatedly… • … until it is in independent arrays / elements e.g. 2, 18, 6, 4, 12, 3 • Compare the first two items (index 0 and 1) e.g. 2, 18 • … and combine to create a new array in descending order i.e. 18, 2 • Repeat with indexes 2 and 3 (6, 4), then 4 and 5 (12, 3) • Compare the first element in the first two new arrays • …Choose the largest element, writing this to the new array first • …repeat until no elements left • Combine the two remaining lists into one list e.g. [2, 18, 6, 4, 12, 3] [2, 18, 6] [4, 12, 3] [2, 18] [6] [4, 12] [3] [2] [18] [6] [4] [12] [3] [18, 2] [6, 4] [12, 3] [18, 6, 4, 2] [12, 3] [18, 12, 6, 4, 3, 2] e.g. [2, 18, 6, 4, 12, 3] [2, 18, 6] [4, 12, 3] [2, 18] [6] [4, 12] [3] [2] [18] [6] [4] [12] [3] [18 2] [6] [12 4] [3] [18 6 2] [12 4 3] [18, 12, 6, 4, 3, 2]	6 **AO1.2 (3)** **AO2.2 (3)**	Allow max 5 if correct description but in ascending order.
5b	1 mark per bullet • Merge sort might create a new array each time it splits and merges / often implemented recursively which places additional data on the stack • Insertion sort does not use any additional arrays//Insertion sort is an in-place algorithm.	2 **AO1.2 (2)**	

6	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of data mining; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate is able to weigh up the context which results in a supported and realistic judgment as to whether it is possible to use data mining in this context. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knowledge and understanding of data mining; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether it is possible to use data mining. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence.* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of data mining with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** • Extracting data from databases • Using large data sets • Looking for patterns/specific occurrences of data • Gathering data that can be analysed and used to inform decisions **AO2: Application** e.g. • Use to find out what his users do • Find features that are used most often • Find features that are not used • Find out what people in his target age group do on other sites • Find out characteristics of people who use the site **AO3: Evaluation** e.g. • Can identify areas to focus attention • Save time and money by identifying areas that are not popular/used • New features targeted at specific groups could bring in new business e.g. advertising • But care would need to be applied to privacy issues / GDPR and potential impact on the users

7ai	1 mark per example e.g. • No actual images shown • Items are named / labelled • Simplified layout with shapes	2 **AO2.1 (2)**	Allow any reasonable examples, but they must be for different aspects
7aii	1 mark per bullet to max 3 e.g. • Reduces complexity of design • Reduces complexity of programming • Reduce memory/processing requirements • Could involve a large number of images that would take excessive memory • Reality contains things that aren't relevant to a computer program	3 **AO1.1 (1)** **AO1.2 (1)** **AO2.1 (1)**	Note: do not allow answers related to the user experience / user interpretation, the question is about the production of the system
7aiii	1 mark per example e.g. • Garden dimensions/width/length • Number of items in the garden • Name of items in the garden • Location of items in the garden	3 **AO2.1 (3)**	

7bi	1 mark per bullet to max 4
	• Class declaration
	• 3 attributes declared
	• Constructor
	• …taking parameters
	• …setting the attributes to the parameters
	
	e.g.
	`class GardenItem`
	`  private itemName`
	`  private length`
	`  private width`
	`  public procedure new(pItemName, pLength, pWidth)`
	`  itemName = pItemName`
	`  length = pLength`
	`  width = pWidth`
	`  endprocedure`
	`endclass`
		4
		
		AO1.1 (1)
		AO2.1 (1)
		AO2.2 (1)
		AO3.2 (1)
			Note that example answers are given in the specification pseudocode. Any pseudocode answer that can be understood by a 'competent programmer' should be accepted

<table>
<tr>
<td>7bii</td>
<td>1 mark per bullet
• Class declaration inheriting from <code>gardenItem</code>
• Additional 3 properties declared as private
• Constructor takes all 5 parameters
• Use of super (or equivalent) to set super class parameters
• Remainder of properties set to parameters

e.g.
<pre>class Tree inherits GardenItem
  private height
  private sun
  private shade
  public procedure new(pName, pHeight, pLenWidth,
pSun, pShade)
    super.itemName = pName
    super.length = pLenWidth
    super.width = pLenWidth
    height = pHeight
    sun = pSun
    shade = pShade
  endprocedure
endclass</pre></td>
<td>5

AO1.1 (1)
AO2.2 (1)
AO3.2 (3)</td>
<td>Accept solutions that call the parent’s constructor.

<pre>class Tree inherits GardenItem
  private height
  private sun
  private shade
  public procedure new(pName,
pHeight, pLenWidth, pSun, pShade)
    height = pHeight
    sun = pSun
    shade = pShade

super.new(pName,pLenWidth,pLenWidth)
  endprocedure
endclass</pre></td>
</tr>
<tr>
<td>7biii</td>
<td>1 mark per bullet
• Declaration of instance of tree (i.e. <code>new Tree</code>)
• Storing result in <code>firstTree</code>
• All parameters included and in the same order as 7bii
• …with appropriate data types

e.g.
<pre>firstTree = new Tree("Common Oak",40,40,true, true)</pre></td>
<td>4

AO1.1 (1)
AO2.2 (1)
AO3.2 (2)</td>
<td></td>
</tr>
</table>

7biv	1 mark per bullet
	• Get method declaration
	• Returns `itemName`
	• Set method declaration
	• …takes value as a parameter
	• …assigns parameter to `itemName`
	e.g.
	`function getItemName()`
	`  return itemName`
	`endfunction`
	
	`procedure setItemName(newname)`
	`  itemName = newname`
	`endprocedure`
		4
		
		AO1.2 (2)
		AO2.2 (2)	

7bv	1 mark per bullet to max 6
	• Procedure declaration with all four parameters
	• Looping 1000 times / to end of array
	• Checking if height and width are less than or equal to the maximum height and width
	• Checking if sun and shade match
	• Outputting value(s) using get methods
	• Output an appropriate message
	• Outputs a message if no matching tree is found
	e.g.
	`procedure findTree(pHeight, pWidth, pSun, pShade)`
	`  flag = false`
	`  for i = 0 to 999`
	`  if treeArray[i].getHeight() <= pHeight then`
	`  if treeArray[i].getWidth() <= pWidth then`
	`  if treeArray[i].getSun() == pSun then`
	`  if treeArray[i].getShade() == pShade then`
	`  flag = true`
	`  print(treeArray[i].getItemName() + " height: " + treeArray[i].getHeight + " width: " + treeArray[i].getWidth() + " Sun?: " + treeArray[i].getSun() + " Shade?: " + treeArray[i].getShade()`
	`  endif`
	`  endif`
	`  endif`
	`  endif`
	`  next count`
	`  if flag == false then`
	`  print("No suitable trees")`
	`  endif`
	`endprocedure`
		6
		
		AO1.2 (1)
		AO2.2 (2)
		AO3.2 (3)	

7c	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of caching and reuseable components; the material is generally accurate and detailed. The candidate is able to apply their knowledge and understanding directly and consistently to the context provided. Evidence/examples will be explicitly relevant to the explanation. The candidate is able to weigh up the use of both caching and reusable components which results in a supported and realistic judgment as to whether it is possible to use them in this context. *There is a well-developed line of reasoning which is clear and logically structured. The information presented is relevant and substantiated.* **Mark Band 2 – Mid level** **(4-6 marks)** The candidate demonstrates reasonable knoledge and understanding of caching and reuseable components; the material is generally accurate but at times underdeveloped. The candidate is able to apply their knowledge and understanding directly to the context provided although one or two opportunities are missed. Evidence/examples are for the most part implicitly relevant to the explanation. The candidate makes a reasonable attempt to come to a conclusion showing some recognition of influencing factors that would determine whether it is possible to use caching and reusable componments in this context. *There is a line of reasoning presented with some structure. The information presented is in the most part relevant and supported by some evidence* **Mark Band 1 – Low Level** **(1-3 marks)** The candidate demonstrates a basic knowledge of caching and reuseable components with limited understanding shown; the material is basic and contains some inaccuracies. The candidates makes a limited attempt to apply acquired knowledge and understanding to the context provided. The candidate provides nothing more than an unsupported assertion. *The information is basic and comunicated in an unstructured way. The information is supported by limited evidence and the relationship to the evidence may not be clear.* **0 marks** No attempt to answer the question or response is not worthy of credit.	**9** **AO1.1 (2)** **AO1.2 (2)** **AO2.1 (2)** **AO3.3 (3)**	**AO1: Knowledge and Understanding** **Indicative content** Caching: • Data that has been used is stored in cache/RAM in case it is needed again • Allows faster access for future use Reusable components • One piece of code can be used in multiple places / called many times • Use of subroutines / procedures / functions • Use of classes • Use of external libraries **AO2: Application** • Store items in cache • Store requirements in cache • Store garden layout in cache • Reuse shapes / designs • The use of a class allows replication **AO3: Evaluation** e.g. • Faster development • Faster/easier future adaptation • Better performance of program • Takes more time to plan/design to make use of both

OCR (Oxford Cambridge and RSA Examinations)
The Triangle Building
Shaftesbury Road
Cambridge
CB2 8EA

OCR Customer Contact Centre

Education and Learning
Telephone: 01223 553998
Facsimile: 01223 552627
Email: general.qualifications@ocr.org.uk

www.ocr.org.uk

For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored

Oxford Cambridge and RSA Examinations
is a Company Limited by Guarantee
Registered in England
Registered Office; The Triangle Building, Shaftesbury Road, Cambridge, CB2 8EA
Registered Company Number: 3484466
OCR is an exempt Charity

OCR (Oxford Cambridge and RSA Examinations)
Head office
Telephone: 01223 552552
Facsimile: 01223 552553

© OCR 2019