GCE

Computer Science

H446/02: Algorithms and programming

A Level

Mark Scheme for June 2024

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.

MARKING INSTRUCTIONS

PREPARATION FOR MARKING

1. Make sure that you have accessed and completed the relevant training packages for on-screen marking: *RM assessor Online Training*; *OCR Essential Guide to Marking*.
2. Make sure that you have read and understood the mark scheme and the question paper for this unit. These are posted on the RM Cambridge Assessment Support Portal http://www.rm.com/support/ca
3. Log-in to RM assessor and mark the **required number** of practice responses ("scripts") and the **number of required** standardisation responses.

YOU MUST MARK 10 PRACTICE AND 10 STANDARDISATION RESPONSES BEFORE YOU CAN BE APPROVED TO MARK LIVE SCRIPTS.

MARKING

1. Mark strictly to the mark scheme.

2. Marks awarded must relate directly to the marking criteria.

3. The schedule of dates is very important. It is essential that you meet the 50% and 100% deadlines. If you experience problems, you must contact your Team Leader (Supervisor) without delay.

4. If you are in any doubt about applying the mark scheme, consult your Team Leader by telephone or the RM assessor messaging system, or by email.

5. **Crossed Out Responses**
   Where a candidate has crossed out a response and provided a clear alternative then the crossed-out response is not marked. Where no alternative response has been provided, examiners may give candidates the benefit of the doubt and mark the crossed-out response where legible.

   Contradictory Responses

   When a candidate provides contradictory responses, then no mark should be awarded, even if one of the answers is correct.

   Short Answer Questions (requiring only a list by way of a response, usually worth only **one mark per response**)

   Where candidates are required to provide a set number of short answer responses then only the set number of responses should be marked. The response

space should be marked from left to right on each line and then line by line until the required number of responses have been considered. The remaining responses should not then be marked. Examiners will have to apply judgement as to whether a 'second response' on a line is a development of the 'first response', rather than a separate, discrete response. *(The underlying assumption is that the candidate is attempting to hedge their bets and therefore getting undue benefit rather than engaging with the question and giving the most relevant/correct responses.)*

Short Answer Questions (requiring a more developed response, worth **two or more marks**)

If the candidates are required to provide a description of, say, three items or factors and four items or factors are provided, then mark on a similar basis – that is downwards (as it is unlikely in this situation that a candidate will provide more than one response in each section of the response space.)

Longer Answer Questions (requiring a developed response)

Where candidates have provided two (or more) responses to a medium or high tariff question which only required a single (developed) response and not crossed out the first response, then only the first response should be marked. Examiners will need to apply professional judgement as to whether the second (or a subsequent) response is a 'new start' or simply a poorly expressed continuation of the first response.

6. Always check the pages (and additional objects if present) at the end of the response in case any answers have been continued there. If the candidate has continued an answer there then add a tick to confirm that the work has been seen.

7. Award No Response (NR) if:

- there is nothing written in the answer space

Award Zero '0' if:

- anything is written in the answer space and is not worthy of credit (this includes text and symbols).

Team Leaders must confirm the correct use of the NR button with their markers before live marking commences and should check this when reviewing scripts.

8. The RM assessor **comments box** is used by your team leader to explain the marking of the practice responses. Please refer to these comments when checking your practice responses. **Do not use the comments box for any other reason.**

If you have any questions or comments for your team leader, use the phone, the RM assessor messaging system, or e-mail.

9. *Assistant Examiners will send a brief report on the performance of candidates to their Team Leader (Supervisor) via email by the end of the marking period. The report should contain notes on particular strengths displayed as well as common errors or weaknesses. Constructive criticism of the question paper/mark scheme is also appreciated.*

10. For answers marked by levels of response:

a. **To determine the level** – start at the highest level and work down until you reach the level that matches the answer
b. **To determine the mark within the level**, consider the following:

Descriptor	**Award mark**
On the borderline of this level and the one below	At bottom of level
Just enough achievement on balance for this level	Above bottom and either below middle or at middle of level (depending on number of marks available)
Meets the criteria but with some slight inconsistency	Above middle and either below top of level or at middle of level (depending on number of marks available)
Consistently meets the criteria for this level	At top of level

11. 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)
	Blank Page – this annotation must be used on all blank pages within an answer booklet (structured or unstructured) and on each page of an additional object where there is no candidate response.
	Level 1
	Level 2
	Level 3

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>1ai</td><td>1 mark each
• A memory location // named storage location
• Stores/holds data / a value
• That can be changed</td><td>2</td><td>Element/identifier on its own is not enough for BP1</td></tr>
<tr><td>1aii</td><td>1 mark for all variables:
a b c</td><td>1</td><td>Must have all three for the mark to be awarded.</td></tr>
<tr><td>1b</td><td>1 mark for do loop and 1 mark for while loop up to a maximum of 2 marks.
• While loop will check the condition at the start of the loop // pre-condition loop
• Do loop will check the condition at the end of the loop // post-condition

• The code in a while loop may never run (if the condition is already met)
• The code in a do loop will always run at least one.</td><td>2</td><td>Answer must cover both do loop and while loop for 2 marks to be awarded

BP1 and BP2 must be specific as to the location that the condition is placed</td></tr>
<tr><td>1c</td><td>1 mark each
• Initialisation of counter variable before loop
• Condition (e.g. while c <= a)
• c incremented in loop
• Completed loop will produce correct results 12, 24 … 144

e.g.
<pre>c = 1
while c <= a
   print(c * a)
   c = c + 1
endwhile</pre><pre>c = 0
while c < a
   c = c + 1
   print(c * a)
endwhile</pre><pre>c = 0
while c <> a
   c = c + 1
   print(c * a)
endwhile</pre></td><td>4</td><td>BP2 – Allow any suitable logic for the while loop condition that iterates between 1 and a.

Allow != for <>
Allow += or equivalent for c = c + 1

Allow hard coded values for upper bound such as a = 12 or a = 13 depending on the relational operator used.

No marks awarded if a conditional loop has not been used.

Max 3 if solution does not completely work.</td></tr>
</table>

2	**Mark Band 3 – High level** **(9-12 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. *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** **(5-8 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 provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. *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** **(1-4 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 a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. *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.	**12**	**AO1: Knowledge and Understanding** **Indicative content** • Analysis of patterns and anomalies in large data sets • Turns large quantities of data into useful information. These may not be immediately obvious to a casual reader. • Resulting information is used to make predictions, to increase revenue, to target advertising and improve services. **AO2: Application** • Identify the amount of time students spend on the system • Identify the days / times when is it used most? Least? • Identify the features/tools students use most and least • Identify which questions find the most difficult. • Identify the time / day of the week when students learn the most • Identify which schools are performing better or which areas of the country are performing better. • Identify which courses students are enjoying more **AO3: Evaluation** • Data mining can spot patterns/trends, however it cannot explain them. Company may still need to do extra research. • Students may have privacy concerns about their activities being logged. • Requires powerful computers with a lot of processing power to process huge amounts of data. • Need to make sure data collection is legal and in terms and conditions • Analysis can be costly e.g. may need an external company • Security - data being collected requires protection

<table>
<tr><td></td><td></td><td></td><td>

- Include more features students use so they will use it more, and gain more revenue
- Identifying features not used means resources and development can be moved from these onto other areas

</td></tr>
<tr><td>3a</td><td>

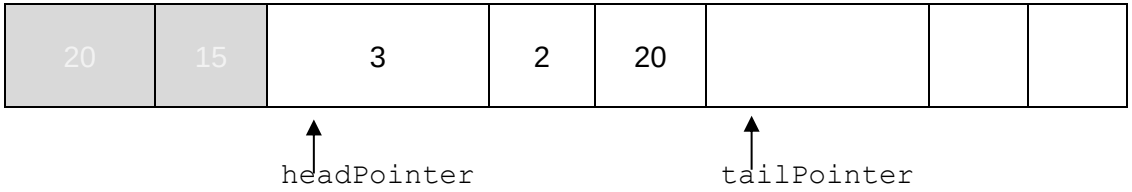
<table>
<tr><th>Node</th><th>Distance travelled</th><th>Heuristic</th><th>Distance travelled + Heuristic</th><th>Previous node</th><th>Marking Guidance</th></tr>
<tr><td>A</td><td>0</td><td>90</td><td>90</td><td>N/A</td><td>1 Mark</td></tr>
<tr><td>B</td><td>20</td><td>80</td><td>100</td><td>A</td><td rowspan="2">1 Mark</td></tr>
<tr><td>C</td><td>44</td><td>43</td><td>87</td><td>A</td></tr>
<tr><td>D</td><td>128</td><td>70</td><td>198</td><td>E</td><td>1 Mark</td></tr>
<tr><td>E</td><td>66</td><td>20</td><td>86</td><td>C</td><td>1 Mark</td></tr>
<tr><td>F</td><td>81</td><td>8</td><td>89</td><td>E</td><td>1 Mark</td></tr>
<tr><td>G</td><td>90</td><td>0</td><td>90</td><td>F</td><td>1 Mark</td></tr>
</table>

Path: A→ C →E → F → G Distance: 90 (1 Mark)

</td><td>7</td><td>For Row A allow N/A, None, Null, - or a blank cell / equivalent.</td></tr>
</table>

3b	1 mark for each difference up to a maximum of 4 marks: e.g. • Trees have one **root** node // graphs do not have a root node (1) • Trees do not allow **cycles**/loops // graphs do allow cycles / loops (1) • Trees store **hierarchy** // graphs have no hierarchy (1) • Trees are always **undirected** // graphs can be directed (1) • Trees are always **connected** // graphs can be connected or disconnected (1)	**4**	Do not allow responses related to weighted / unweighted.

3c	**Mark Band 3 – High level (7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of heuristics; 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. 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 heuristics; 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 provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed. 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 heuristics 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 a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated. 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.	**9**	**AO1: Knowledge and Understanding** **Indicative content** • Heuristics are used to reduce time taken to solve a problem • It is a general 'rule of thumb' or an educated guess. • It finds a solution which is 'good enough' / close to the best solution • Heuristic is a weight added to a node/decision • E.g. Description of use such as in A* algorithm as estimate of distance to destination **AO2: Application** • Heuristics reduce the time complexity as every possibilities within the game does not need to be examined. • Heuristics require skill to implement effectively • Used in AI when the exact steps cannot be pre-programmed and decisions need making • Due to time-saving, they are not always accurate, the solution e.g. shortest path might not be the most efficient. **AO3: Evaluation** • Heuristics are more appropriate with complex time-critical tasks - some aspects of game may require faster searching/decisions - current graph is not complex or time-critical so not required • Heuristics are more appropriate with large-scale tasks - game could be large scale and AI algorithms may need to be shortened • Games are not life-critical, so a good answer is likely enough, a perfect answer is not necessarily required. • Avoid programs running indefinitely - in a computer game there could be too many

	0 marks No attempt to answer the question or response is not worthy of credit.		possibilities so will terminate with a solution faster
4a	1 mark each • `headPointer` to identify the **first item/element** in the queue // identify which item to dequeue/remove next • `tailpointer` to identify the **next free** space in the queue // identify where the next item/element will be enqueued/added	**2**	
4b	1 mark for queue elements 1 mark for both pointers 	**2**	Allow 20 and 15 in place but crossed out OR allow 20 and 15 in place only if `headPointer` and `tailPointer` are correct
4c	1 mark each to max 2 e.g. • A queue is a FIFO structure // elements processed in the order entered • A queue will not allow new data inserted at the front // only allows new data to be enqueued at the rear • The queue contents cannot be resequenced/sorted without rewriting	**2**	

<table>
<tr>
<td>5a</td>
<td>1 mark each
• Check if the stack is empty // check <code>topStack</code> is equal to 0
• …and if so return a suitable value (e.g. -1/ null) // do nothing //give warning
• (If not empty) decrement <code>topStack</code>
• Return the value in element <code>topStack</code> from the array numbers</td>
<td>4</td>
<td>Do not award BP3 if a value has been returned from the function for BP4 first.</td>
</tr>
<tr>
<td>5b</td>
<td>1 mark for each completed statement

<pre>function push(dataValue)
    if topStack != 100 then
        numbers[topStack] = dataValue
        topStack = topStack + 1
        return true
    else
        return false
    endif
end function</pre></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5c</td>
<td>1 mark each
• Calling push() with parameter 15
• … storing/using return value in selection …
• … comparing true/false (may be implicit e.g. <code>if push(15) then</code>) …
• … outputting a suitable message if false and if true

e.g.

<pre>added = push(15)
if added = false then
  print("Not added")
else
  print("Added")
endif</pre>
<pre>if push(15) then
   print("Added")
else
  print("Not Added")
endif</pre></td>
<td>4</td>
<td>True/False comparisons must be Boolean values and not strings, but allow FT after that.

If <code>push()</code> is called twice BP4 cannot be awarded.</td>
</tr>
</table>

<table>
<tr><td>6a</td><td>1 mark each
Compare the first element (rainbow) to search item / clouds
If it is equal to the search item return index / found
If it is not equal move to the next element
Repeat until either search item / clouds is equal // or the end of the list has been reached</td><td>3</td><td>Allow answers by example from the given dataset</td></tr>
<tr><td>6b</td><td>1 mark for: the data is not in order/sorted</td><td>1</td><td></td></tr>
<tr><td>6c</td><td>
<table>
<tr><th>rainbow</th><th>moon</th><th>sun</th><th>stars</th><th>clouds</th><th>tornado</th><th>Marking Guidance</th><th>Marking Guidance</th></tr>
<tr><td>rainbow</td><td>moon</td><td>sun</td><td>stars</td><td>clouds</td><td>tornado</td><td></td><td></td></tr>
<tr><td>moon</td><td>rainbow</td><td>sun</td><td>stars</td><td>clouds</td><td>tornado</td><td>Values change</td><td>1 Mark</td></tr>
<tr><td>moon</td><td>rainbow</td><td>sun</td><td>stars</td><td>clouds</td><td>tornado</td><td></td><td>1 Mark</td></tr>
<tr><td>moon</td><td>rainbow</td><td>stars</td><td>sun</td><td>clouds</td><td>tornado</td><td>Values change</td><td>1 Mark</td></tr>
<tr><td>clouds</td><td>moon</td><td>rainbow</td><td>stars</td><td>sun</td><td>tornado</td><td>Values change</td><td>1 Mark</td></tr>
<tr><td>clouds</td><td>moon</td><td>rainbow</td><td>stars</td><td>sun</td><td>tornado</td><td></td><td>1 Mark</td></tr>
</table>
</td><td>5</td><td>If candidate has given <u>descending</u> order, max 4.

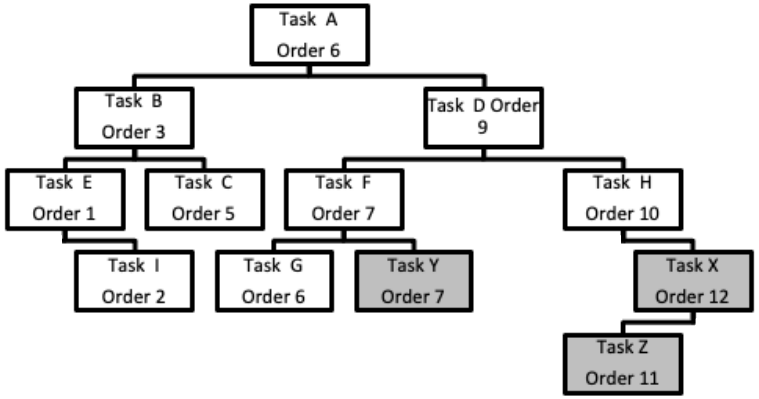
MP1, MP3 and MP4 are lines that show a change of values during a pass.

MP2 and MP5 do not have to be explicitly given in full if there is a comment to identify no change occur during the pass.

Award no marks if not an insertion sort.</td></tr>
<tr><td>6di</td><td>1 mark each
<u>Linear</u>
The time will (increase) in direct proportion to the number of items</td><td>2</td><td></td></tr>
<tr><td>6dii</td><td>1 mark each
<u>Logarithmic</u>
The additional memory space required grows at a decreasing rate as the number of items increases</td><td>2</td><td></td></tr>
<tr><td>6diii</td><td>Constant // O(1)</td><td>1</td><td></td></tr>
<tr><td>6div</td><td>Exponential // $O(2^n)$ // $O(K^n)$</td><td>1</td><td></td></tr>
</table>

7a	1 mark each to max 2 for justification e.g. • Can store multiple items of data under **one identifier** // so all the data about a task can be accessed using the same identifier • Can store data of **different data types** and this task has string, real and integers	**2**	
7bi	1 mark each e.g. • Each node can have 0, 1 or 2 child nodes // a maximum of 2 child nodes • Nodes are ordered (with left nodes less than the parent and right nodes greater) • The location to which a node is added depends on its order.	**2**	
7bii	1 mark for advantage e.g. • Searching is faster (O(log n)) • Inserting new tasks is faster • Do not need to sort the structure (each time a new task is inserted)	**1**	

<table>
<tr>
<td>7biii</td>
<td>1 mark for each row
<table>
<tr><th>Statement</th><th>Depth-first (post-order)</th><th>Breadth-first</th><th>Neither of these two traversals</th></tr>
<tr><td>All nodes at the current depth are visited before moving to the next depth</td><td></td><td>✓</td><td></td></tr>
<tr><td>The algorithm traverses to the end of one branch before moving to another branch</td><td>✓</td><td></td><td></td></tr>
<tr><td>The algorithm will make use of backtracking</td><td>✓</td><td></td><td></td></tr>
<tr><td>The traversal can be used to output the contents of the tree in ascending order</td><td></td><td></td><td>✓</td></tr>
<tr><td>The algorithm will output the root node last</td><td>✓</td><td></td><td></td></tr>
</table>
</td>
<td>5</td>
<td></td>
</tr>
</table>

7biv	1 mark for Task Y to right of Task F 1 mark for Task X to right of Task H … … 1 mark for Task Z to left of Task X  	**3**	The direction of left/right child nodes must be clear and cannot just be a downward vertical line.
8a	1 mark for each input to max 2 e.g. • Entering a name • Selecting a vehicle • Pressing arrow key to move forward • Pressing arrow key to move backward • Pressing arrow key to move left • Pressing arrow key to move right 1 mark for each output to max 2 e.g. • Images of vehicles to choose from • Background of area • Image of other vehicles • Image of controls and description of what they do	**4**	Allow any feasible input/output for scenario

8bi	1 mark for definition, 1 mark each for each example of use to max 2 (3 overall) **Definition:** • Removal of unnecessary detail **Example use:** • E.g. simplifying scenery • E.g. removing internal features of a vehicle that are not needed • E.g. simplify physics for vehicle movement • E.g. The vehicles may not be drawn to scale	**3**	Allow any reasonable examples for this scenario For the example use, allow 2 marks for stating a valid example of abstraction with an expansion. For example, "simplify track (1) by taking out the bumps in the road (1)" would be given two marks.
8bii	1 mark each to max 2 • Simplifies the problem / algorithm / programming code • Faster to create the program code • Final program uses less memory/processor time • Programmer can focus on core aspects of the game • Completed game will be simpler for end users to understand / play	**3**	Do not accept a reiteration of a definition of abstraction.
8ci	Splitting the problem down into smaller (sub) problems	**1**	
8cii	1 mark each to max 2 e.g. • To break the problem down into individual components • …to see which components can be tackled concurrently • Identify any resusable program elements … • …to avoid creating the same algorithm twice • Split the program between individuals • … so they can focus on individual elements // to focus on their speciality • Identify the subroutines and how they will interract • …so everyone knows the requirements for their part of the problem • Easier to tackle/focus on one smaller problem at a time… • … so this simplifies writing/testing code	**2**	Allow for other valid benefits of using decomposition.

<table>
<tr>
<td>9a</td>
<td>1 mark each to max 3. Max 2 for generic answers with no relation to scenario. e.g.
• Has a set/fixed number of values
• ... and the number of spaces in the road will not change
• Stores data of one type
• … as the array is only made up of prize objects
• Stores data linearly
• … match the linear nature of the road
• Array contents are mutable
• … so prizes can be added/removed from the road
• A single indentifier is used to directly index
• … any position in the road
• Can be iterated by index
• … to perform an operation on all road positions</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>9bi</td>
<td>1 mark each
• Function/subroutine with identifier <code>getName</code> taking no parameters
• Returning <code>name</code>

e.g.

<pre>public function getName()
   return name
endfunction</pre>
<pre>def getName(self):
   return self.__name</pre>
<pre>public getName(){
   return name
}</pre>
<pre>function getName() {
   return this.name
}</pre></td>
<td>2</td>
<td>BP1 Do not award procedure or method

BP1 Allow self as an additional parameter if Python is used.

BP1 If an access modifier is given for the method, it must be public and not private.

BP2 Do not allow any modified name attribute to be returned.</td>
</tr>
</table>

9bii	1 mark each
	• New instance of `prize …`
	• … with “Box”, “money” and 25 as parameters
	• Assigned to `allPrizes` index 3
	
	e.g.
	`allPrizes[3] = new prize("Box", "money", 25)`
	
	`allPrizes[3] = prize.new("Box", "money", 25)`
	
	`allPrizes[3] = prize("Box", "money", 25)`
		3	MP2 allow any order of parameters
			
			`"Box"` and `"Money"` must be strings and `25` must be an integer
			
			Allow `prize.new()` as new is given as the constructor method in the class diagram

9biii	1 mark for each bullet to maximum 3 e.g. • **Decision** - check whether the space already has a prize allocated … • **Action if true** - another space/number will need to be generated • **Action if false** - the prize will be stored here • **Decision** - check if all 10 prizes have been allocated … • **Action if true** - the algorithm needs to stop generating numbers • **Action if false** - a new number/space needs to be generated and checked	**3**	Give: • 1 mark for stating a decision • 1 mark for the action required if true • 1 mark for the action required if false

<table>
<tr>
<td>9ci</td>
<td>1 mark each

• Constructor header (any suitable name e.g. new, constructor, create, init)
• …taking one parameter only
• Initialising name to the parameter
• Initialising money to 5
• Initialising experience to 0 and roadPosition to 0

e.g.

Pseudocode example:
<pre>public procedure new(pName)
    name = pName
    experience = 0
    roadPosition = 0
    money = 5
endprocedure</pre>
Python Example:
<pre>def __init__(self, pName):
    self.__name = pName
    self.__experience = 0
    self.__roadPosition = 0
    self.__money = 5</pre>
C# Example:
<pre>public Character(string pName) {
     string name = pName;
     int experience = 0;
     int roadPosition = 0;
     int money = 5;
}</pre></td>
<td>5</td>
<td>Allow minor changes to identifiers as long as purpose is clear.

Allow
<pre>procedure new(pName)
    this.name = pName
    …</pre>
(or similar e.g. self.name)

Allow two parameters if one is self and the response is clearly in Python.

The parameter name should be different to the attribute name.</td>
</tr>
</table>

<table>
<tr>
<td>9cii</td>
<td>1 mark each
• Procedure/method header …
• … taking two parameters, type (or similar) followed by value (or similar) …
• … compare type parameter with "money"
• … compare type parameter with "experience"
• … both attributes updated correctly and nothing else modified

e.g.
<pre>
public procedure updateValues(pType, pValue)
   if pType == "money" then
      money = money + pValue
   elseif pType == "experience"
      experience = experience + pValue
   endif
endprocedure

def updateValues(self, pType, pValue):
   if pType == "money":
      money += pValue
   elif pType == "experience":
      experience += pValue
</pre>
</td>
<td>5</td>
<td>Do not allow Function for BP1

BP2 parameters must be given in the correct order to match the calls to updateValues() in the question.

"money" and "experience" must be string values</td>
</tr>
</table>

9d	1 mark for each completed space
	
	`character1 = new Character("Jamal")`
	`newPosition = 0`
	`while newPosition < 50`
	`  move = random(1, 4)`
	`  character1.changePosition(move)`
	`  newPosition = character1.getRoadPosition()`
	`  if newPosition < 50 and road[newPosition] != null then`
	`  prizeType = road[newPosition].getType()`
	`  valueAmount = road[newPosition].getValue()`
	`  character1.updateValues(prizeType, valueAmount)`
	`  print("Congratulations you are in position", newPosition, "and found", road[newPosition].getName())`
	`  print("Money", character1.getMoney(), "and experience", character1.getExperience())`
	`  endif`
	`endwhile`
	`print("You reached the end of the road")`
		6	Allow `road.length` // `len(road)` instead of `50`
			
			Allow `<=49` instead of `< 50`

9e	1 mark each
	• (Line 02)
	`for x = 0 to 49`
	• (Line 03)
	`print("Space", x)`
	• (Line 06)
	`else // elseif road[x] <> null`
	• (Line 07)
	`print(road[x].getName())`
		4	Line `07` allow `print(road[x].name)`

9f	**Mark Band 3 – High level** **(7-9 marks)** The candidate demonstrates a thorough knowledge and understanding of global variables and the alternatives; 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.*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 global variables and the alternatives; 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 provides a reasonable discussion, the majority of which is focused. Evaluative comments are, for the most part appropriate, although one or two opportunities for development are missed.*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 global variables and the alternatives 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 a limited discussion which is narrow in focus. Judgements if made are weak and unsubstantiated.*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.*	**9**	**AO1: Knowledge and Understanding** **Indicative content** • Global variables are created when the program starts, all subroutines can access/update the contents • Local variables are created in the subroutine they are created in, they are not accessible directly from any other subroutine • Local variables are removed from memory when the subroutine ends. • Local variables can be passed as parameters to a function to be updated, and then returned to override the original local variable • Local variables can be passed by reference to a subroutine to allow the content of the variable to be updated **AO2: Application** • The variables will be stored in memory throughout the whole code execution. However, the amount of data they are storing is relatively low so would not use a lot of memory. • When the game is expanded, the amount of data may increase so it could be memory intensive, especially if graphics are used in the game. • Both arrays are needed throughout the whole game so keeping them as global will make writing the code easier as the programmer will not need to keep passing them as parameters and setting return values. • Only one part of the game is being created initially and therefore the use of global variables would not affect the efficiency greatly. However, when the program expands, it could cause accuracy / testing / debugging and maintenance problems. **AO3: Evaluation** • As this is only a prototype, the use of global variables would be beneficial. • However, when the game expands, the use of global variables could create issues such as running out of memory, coupling, testing & debugging problems and maintenance problems. • The programmer may be best to keep the variables as local and then pass them between the different subroutines as parameters byVal and byRef.

	0 marks No attempt to answer the question or response is not worthy of credit.		

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our customer support centre.

Call us on

01223 553998

Alternatively, you can email us on

support@ocr.org.uk

For more information visit

ocr.org.uk/qualifications/resource-finder

ocr.org.uk

Twitter/ocrexams

/ocrexams

/company/ocr

/ocrexams

CAMBRIDGE
UNIVERSITY PRESS & ASSESSMENT

OCR is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

OCR operates academic and vocational qualifications regulated by Ofqual, Qualifications Wales and CCEA as listed in their qualifications registers including A Levels, GCSEs, Cambridge Technicals and Cambridge Nationals.

OCR provides resources to help you deliver our qualifications. These resources do not represent any particular teaching method we expect you to use. We update our resources regularly and aim to make sure content is accurate but please check the OCR website so that you have the most up-to-date version. OCR cannot be held responsible for any errors or omissions in these resources.

Though we make every effort to check our resources, there may be contradictions between published support and the specification, so it is important that you always use information in the latest specification. We indicate any specification changes within the document itself, change the version number and provide a summary of the changes. If you do notice a discrepancy between the specification and a resource, please contact us.

Whether you already offer OCR qualifications, are new to OCR or are thinking about switching, you can request more information using our Expression of Interest form.

Please get in touch if you want to discuss the accessibility of resources we offer to support you in delivering our qualifications.