



# Cambridge IGCSE™

---

**COMPUTER SCIENCE****0478/22**

Paper 2

**May/June 2020**

MARK SCHEME

Maximum Mark: 50

---

**Published**

Students did not sit exam papers in the June 2020 series due to the Covid-19 global pandemic.

This mark scheme is published to support teachers and students and should be read together with the question paper. It shows the requirements of the exam. The answer column of the mark scheme shows the proposed basis on which Examiners would award marks for this exam. Where appropriate, this column also provides the most likely acceptable alternative responses expected from students. Examiners usually review the mark scheme after they have seen student responses and update the mark scheme if appropriate. In the June series, Examiners were unable to consider the acceptability of alternative responses, as there were no student responses to consider.

Mark schemes should usually be read together with the Principal Examiner Report for Teachers. However, because students did not sit exam papers, there is no Principal Examiner Report for Teachers for the June 2020 series.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the June 2020 series for most Cambridge IGCSE™ and Cambridge International A & AS Level components, and some Cambridge O Level components.

---

This document consists of **8** printed pages.

**Generic Marking Principles**

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

**GENERIC MARKING PRINCIPLE 1:**

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

**GENERIC MARKING PRINCIPLE 2:**

Marks awarded are always **whole marks** (not half marks, or other fractions).

**GENERIC MARKING PRINCIPLE 3:**

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

**GENERIC MARKING PRINCIPLE 4:**

Rules must be applied consistently e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

**GENERIC MARKING PRINCIPLE 5:**

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

**GENERIC MARKING PRINCIPLE 6:**

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Question	Answer	Marks
1(a)	<p>For <b>two</b> variables  <b>One</b> mark for identification of a variable with a suitable data type and <b>one</b> mark for its use</p> <p>Variable            hourArrived  Data type           integer  Use                    to calculate the price based on the number of hours parking</p> <p>Variable            priceToPay  Data type            real  Use                    store result of price calculation</p> <p>Many correct answers, these are examples only.</p>	4
1(b)	<p>For <b>each</b> validation check  <b>One</b> mark for identification of a check and <b>one</b> mark for the justification</p> <p>Day  Check            presence check  Justification    need to enter day to be able to calculate price</p> <p>Hour  Check            range check  Justification    need to check hour is between 8 and 23</p> <p>Many correct answers, these are examples only.</p>	4

Question	Answer	Marks
1(c)	Any <b>four</b> from: <ul style="list-style-type: none"> <li>• All 5 digits input</li> <li>• Each of the first 4 digits is multiplied by a weight based on its position</li> <li>• Specific method of individualising the digits described</li> <li>• Added together divided by 11</li> <li>• Remainder used to calculate correct 5th digit</li> <li>• Checked against 5th digit input</li> <li>• If the same, then valid</li> </ul>	<b>4</b>
1(d)	Any <b>four</b> from: <ul style="list-style-type: none"> <li>• Input amount</li> <li>• Check amount greater than or equal to the price to pay ...</li> <li>• ... True – add amount to daily total</li> <li>• ... False – ask for payment to be re-entered</li> <li>• ... Until valid amount input</li> </ul> Sample answer <pre> PRINT "Please Enter Payment" INPUT amount WHILE amount &lt; priceToPay DO     PRINT "Insufficient payment, please enter new Payment"     INPUT amount ENDWHILE totalPayment ← totalPayment + amount           </pre>	<b>4</b>
1(e)	Explanation Any <b>four</b> from: <ul style="list-style-type: none"> <li>• Add number of hours selected to arrival hour</li> <li>• Check if this is greater than 16</li> <li>• True – calculate price up to 16 ...</li> <li>• ... using current price in force</li> <li>• ... and add in evening charge</li> <li>• False – calculate price as usual</li> </ul> <p>Programming statements can be used but must be explained.</p>	<b>4</b>

Question	Answer	Marks
2	Real Integer Char/String String Boolean	5

Question	Answer	Marks
3(a)	<b>One</b> mark for error and correction Line 1 <code>HighestMark ← 0</code> Line 7 <code>INPUT Mark[Count]</code> Line 10 <code>HighestMarkStudents ← HighestMarkStudents + 1</code> Line 14 <code>HighestMark ← Mark[Count]</code>	4
3(b)	Any <b>six</b> from: Add variable <code>LowestMark</code> ... ... Set this to a high value for example 100 Add variable <code>LowestMarkStudents</code> ... ... Set this to zero Check if <code>Mark[Count] = LowestMark</code> ... ... True - <b>add 1 to</b> <code>LowestMarkStudents</code> Check if <code>Mark[Count] &lt; LowestMark</code> ... ... True - <b>set</b> <code>LowestMarkStudenta</code> to 1 <b>and set</b> <code>LowestMark</code> to <code>Mark[Count]</code> Add extra output statement	6

Question	Answer	Marks																												
4(a)	<p><b>One</b> mark for correct input (PointsWon and PointsLost)  <b>One</b> mark for correct calculations (Difference)  <b>One</b> mark for correct output</p> <table border="1" data-bbox="582 371 1693 834"> <thead> <tr> <th>PointsWon</th> <th>PointsLost</th> <th>Difference</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td>5000</td> <td>4474</td> <td>526</td> <td>Keep on trying</td> </tr> <tr> <td>6055</td> <td>2000</td> <td>4055</td> <td>Well done move up</td> </tr> <tr> <td>7900</td> <td>9800</td> <td>-1900</td> <td>Sorry move down</td> </tr> <tr> <td>3000</td> <td>2150</td> <td>850</td> <td>Keep on trying</td> </tr> <tr> <td>-1</td> <td>6700</td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	PointsWon	PointsLost	Difference	OUTPUT	5000	4474	526	Keep on trying	6055	2000	4055	Well done move up	7900	9800	-1900	Sorry move down	3000	2150	850	Keep on trying	-1	6700							<b>3</b>
PointsWon	PointsLost	Difference	OUTPUT																											
5000	4474	526	Keep on trying																											
6055	2000	4055	Well done move up																											
7900	9800	-1900	Sorry move down																											
3000	2150	850	Keep on trying																											
-1	6700																													
4(b)	<p>Any <b>three</b> from:</p> <ul style="list-style-type: none"> <li>• Add extra decision box ...</li> <li>• ... before checking for difference greater than or equal to 1000 // change <code>Is difference &gt;= 1000</code> to <code>&gt;= 1000 and &lt;= 5000</code></li> <li>• Check for difference greater than 5000</li> <li>• Add extra Output 'Fantastic leap up two levels'...</li> <li>• ... before flowline returns to input</li> </ul>	<b>3</b>																												

Question	Answer	Marks
5	<p><b>One</b> mark for explanation of dimension  <b>One</b> mark for explanation of index  <b>One</b> mark for inclusion of an example</p> <p>The dimension is the number of indexes required to access an element.                      The index is the position of the element in an array                      For example A[25] is the 25th element of a one-dimensional array.</p>	3

Question	Answer	Marks																																			
6(a)	<p><b>One</b> mark for field and <b>one</b> mark for reason                      Field Juice code                      Reason only unique identifier</p>	2																																			
6(b)	<table border="1" data-bbox="607 730 1816 1190"> <tr> <td data-bbox="607 730 911 798">Field:</td> <td data-bbox="911 730 1214 798">Fruit 1</td> <td data-bbox="1214 730 1516 798">Fruit 2</td> <td data-bbox="1516 730 1816 798">Size</td> <td data-bbox="1816 730 2083 798">Stock level</td> </tr> <tr> <td data-bbox="607 798 911 865">Table:</td> <td data-bbox="911 798 1214 865">JUICE</td> <td data-bbox="1214 798 1516 865">JUICE</td> <td data-bbox="1516 798 1816 865">JUICE</td> <td data-bbox="1816 798 2083 865">JUICE</td> </tr> <tr> <td data-bbox="607 865 911 932">Sort:</td> <td data-bbox="911 865 1214 932"></td> <td data-bbox="1214 865 1516 932"></td> <td data-bbox="1516 865 1816 932"></td> <td data-bbox="1816 865 2083 932"></td> </tr> <tr> <td data-bbox="607 932 911 999">Show:</td> <td data-bbox="911 932 1214 999"><input type="checkbox"/></td> <td data-bbox="1214 932 1516 999"><input type="checkbox"/></td> <td data-bbox="1516 932 1816 999"><input checked="" type="checkbox"/></td> <td data-bbox="1816 932 2083 999"><input checked="" type="checkbox"/></td> </tr> <tr> <td data-bbox="607 999 911 1066">Criteria:</td> <td data-bbox="911 999 1214 1066">="Apple"</td> <td data-bbox="1214 999 1516 1066">="Apple"</td> <td data-bbox="1516 999 1816 1066"></td> <td data-bbox="1816 999 2083 1066"></td> </tr> <tr> <td data-bbox="607 1066 911 1133">or:</td> <td data-bbox="911 1066 1214 1133"></td> <td data-bbox="1214 1066 1516 1133"></td> <td data-bbox="1516 1066 1816 1133"></td> <td data-bbox="1816 1066 2083 1133"></td> </tr> <tr> <td data-bbox="607 1133 911 1190"></td> <td data-bbox="911 1133 1214 1190"></td> <td data-bbox="1214 1133 1516 1190"></td> <td data-bbox="1516 1133 1816 1190"></td> <td data-bbox="1816 1133 2083 1190"></td> </tr> </table> <p data-bbox="320 1225 768 1257">One mark for each correct column</p>	Field:	Fruit 1	Fruit 2	Size	Stock level	Table:	JUICE	JUICE	JUICE	JUICE	Sort:					Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Criteria:	="Apple"	="Apple"			or:										4
Field:	Fruit 1	Fruit 2	Size	Stock level																																	
Table:	JUICE	JUICE	JUICE	JUICE																																	
Sort:																																					
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>																																	
Criteria:	="Apple"	="Apple"																																			
or:																																					