CAMBRIDGE IGCSE (0478)

Module R: Exam technique

Workbooklet mark scheme · 3 parts · 61 marks

R1 and R4 are GCSE (OCR) only and not on 0478.

Contents

Part	Topic	Marks
R2	How Cambridge 0478 Paper 1 and Paper 2 are structured: mark scheme	20
R3	Command words and what they are asking for: mark scheme	20
R6	The 15-mark scenario question (0478 Paper 2): mark scheme	21
	Total	**61**

MODULE R EXAM TECHNIQUE · R2

How Cambridge 0478 Paper 1 and Paper 2 are structured: mark scheme

20 marks · Cambridge 0478 syllabus (exams 2026-2028, version 6, September 2026): section 2 Assessment overview and Content overview; section 3 (start of Algorithms, programming and logic); section 4 Details of the assessment, Logic gate symbols

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Credit any other valid answer that shows the same understanding. All facts are as stated in the 2026-2028 syllabus (checked November 2026); the 2029-2031 syllabus changes Paper 2.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 s2 Assessment overview; s4 Paper 1, Paper 2, Scenario question; s3 Algorithms, programming and logic (intro)	One mark for each correct row: The exam lasts 1 hour 45 minutes: Both papers The questions are set on Topics 1 to 6: Paper 1 only The last question is a 15-mark unseen scenario question: Paper 2 only Calculators are not allowed: Both papers Coding answers must be in pseudocode (with one exception): Paper 2 only	**5**	A row with more than one box ticked scores 0. Row 5: the exception is the 15-mark scenario question; Paper 1 (Topics 1 to 6) has no coding answers.
2(a) 0478 s2 Assessment overview, Content overview (topics 5, 9, 10)	One mark for each correct row: Explaining how a firewall helps to keep data safe: 1 Completing a truth table for a logic circuit: 2 Stating the output of an SQL query: 2	**3**	Firewalls are Topic 5 (5.3 Cyber security), so Paper 1. Truth tables are Topic 10 (Boolean logic) and SQL is Topic 9 (Databases), so Paper 2. Accept "P1" / "Paper 1" etc.
3(a) 0478 s3 Algorithms, programming and logic (intro); s4 Scenario question	One mark for each correct row: 5-mark question in Cambridge pseudocode: Can be awarded marks 5-mark question in Python: Cannot be awarded marks 15-mark scenario question in Visual Basic: Can be awarded marks 15-mark scenario question in Pascal: Cannot be awarded marks	**4**	A row with both boxes ticked scores 0. Outside the scenario question, solutions written in program code are not awarded marks. In the scenario question only pseudocode, Python, Visual Basic or Java are accepted; any other language is not awarded marks.
4(a) 0478 s2 Assessment overview; s4 Paper 2, Scenario question	One mark for a correct stage of working: 105 − 30 = 75 (minutes left) // 75 − 15 = 60 (marks left) // 75 ÷ 60 One mark for the correct answer: 1.25 (minutes per mark) // 1¼ minutes // 1 minute 15 seconds	**2**	A correct answer with no working scores 2. Working must use 1 hour 45 minutes = 105 minutes and 75 − 15 = 60 marks. Do not credit 1.4 (105 ÷ 75 ignores the 30 minutes and 15 marks of the scenario question) for the answer mark.

QUESTION	ANSWER	MARKS	GUIDANCE
4(b) 0478 s4 Paper 2, Scenario question	5 (minutes)	**1**	Follow through: 4 × their answer to (a). For example, 1.4 in (a) gives 5.6 here, which gains this mark.
5(a) 0478 s3 Algorithms, programming and logic (intro); s4 Logic gate symbols	Any two from: The syllabus says candidates must use the (standard) logic gate symbols (given in section 4 of the syllabus) The AND gate has its own symbol/shape // the AND symbol has a flat (input) side and a curved / rounded (output) side // is D-shaped A box/rectangle with a label written in it is not the AND gate symbol // writing the name of the gate is not enough ... so the drawing would not be credited / would gain no marks (for that gate)	**2**	max 2. Do not credit "it is wrong" or "it is messy" alone. Do not credit answers that say the logic (A AND B) is wrong: the expression shown is correct; only the symbol is wrong.
6(a) 0478 s4 Paper 2, Scenario question	(Answer) A	**1**	Answer A adds 1 to OverCount each time Weight is greater than Limit.
6(b) 0478 s4 Paper 2, Scenario question; s3 Algorithms, programming and logic (intro)	Any two from: Answer B has a logic error // the logic of B is wrong ... it sets/resets OverCount to 1 (each time) instead of adding 1 to it // OverCount can never be more than 1 // the count / output will be wrong (if more than one parcel is over the limit) Answer A only has a syntax error // a missing colon (after the condition) // a small slip in Python syntax Knowledge of programming language syntax is not examined // the logic is more important than the syntax Answer B uses correct pseudocode / syntax but still does not do what the question asks	**2**	max 2. Do not credit "B is wrong" or "A is better" alone. Do not credit "Python is not allowed": Python is allowed in the 15-mark scenario question.

Total: 20 marks

MODULE R EXAM TECHNIQUE · R3

Command words and what they are asking for: mark scheme

20 marks · Cambridge 0478 syllabus command words table, with content from 1.1.1, 1.3.2, 2.1.2(a), 2.2.2, 3.1.2(a), 3.2.1, 5.3.1, 5.3.2, 6.1.2

How to use. One mark per creditworthy point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers for the same point. Words in (brackets) are not needed for the mark. **...** means the point builds on the one before it. Credit any other valid answer that does what the command word asks. Do not credit vague answers such as "faster", "better" or "easier" on their own (too vague).

QUESTION	ANSWER	MARKS	GUIDANCE
1 0478 command words	One mark for each correct row: Define: C Describe: E Explain: A Suggest: D	**4**	B (Compare) is not used. A row with more than one letter scores 0.
2 0478 command words; 1.1.1, 3.2.1, 2.2.2, 3.1.2(a)	One mark for each correct row: Explain why ... binary: answer that gives reasons Give one example ... input device: short answer Describe how ... odd parity check: main features or steps Identify the register ...: short answer	**4**	Two or more ticks in a row = 0 for that row. (Teacher note, the answers: logic gates / switches have two states, so data is processed as 1s and 0s // e.g. keyboard // parity bit makes the number of 1s odd; 1s recounted after transmission; even = error // program counter.)
3(a) 0478 5.3.1; Describe	Any two from: An email / message is sent (to the user / to many people) ... that appears to come from a legitimate / trusted / genuine source (e.g. their bank) It contains a link / an attachment ... the link goes to a fake website The aim is to get the user to give personal / login / bank details (or to download malware)	**2**	max 2. Do not accept answers describing pharming (malicious code that redirects the user without clicking a link). "A scam email" alone = too vague.
3(b) 0478 5.3.2; Explain	Any two from: The URL may be different from / not match the real website's address // may be misspelt / have extra characters ... so staff can see that the link goes to a fake website ... so staff do not click the link / do not enter their details // so personal / login details are not given to the attacker Staff can check the URL by hovering over the link (without clicking it)	**2**	max 2. A reason ("so ...") must be given for the second mark; "check the URL is correct" alone = max 1. Do not accept "the URL stops the email arriving" or answers about firewalls / anti-malware (the question is about checking the URL).
4(a) 0478 2.1.2(a); Compare	Any one from: It only refers to / only describes full-duplex It does not mention half-duplex // gives no similarity or difference between the two methods	**1**	Do not accept "it is wrong": the point about full-duplex is correct.

QUESTION	ANSWER	MARKS	GUIDANCE
4(b) 0478 2.1.2(a)	Any two from: Both send data in both directions Half-duplex sends data in only one direction at a time, but full-duplex sends in both directions at the same time In half-duplex the devices must take turns / wait to send, but in full-duplex they do not need to wait Full-duplex can transfer more data in the same time than half-duplex (for the same connection speed) Example uses that refer to both, e.g. half-duplex: a two-way radio (walkie-talkie) and full-duplex: a telephone call	**2**	max 2. Each point must refer to both methods; a point about one method only (including the given point) = 0. Do not accept "half-duplex only sends data in one direction" (that is simplex) or "full-duplex is faster" alone.
5 0478 6.1.2; Suggest	Any three from, each applied to the farm: Trees are watered only when the soil needs it // the soil is checked all the time / day and night Less water is wasted // lower water costs Fewer workers are needed to water the trees // lower labour costs // staff are free to do other jobs Watering is more consistent / accurate than by hand ... so the trees / fruit grow better // fewer trees are lost from too much or too little water It keeps working when no staff are on the farm (e.g. at night, at weekends) The system reacts more quickly than a person to dry soil	**3**	max 3. Credit any other valid advantage applied to the farm. "Faster", "cheaper" or "more efficient" alone = too vague. The same idea twice scores once.
6 0478 1.3.2; Calculate	One mark for a correct stage of working, one mark for the correct answer: 128 × 64 × 16 (= 131 072 bits) // 131 072 ÷ 8 (= 16 384 bytes) // 16 384 ÷ 1024 16 (KiB)	**2**	max 1 for working. Correct answer with no working seen = 1 mark only (the question asks for all working). Do not accept 16.384 or 16.4 (dividing by 1000): 0478 calculations use 1024.

Total: 20 marks

MODULE R EXAM TECHNIQUE · R6

The 15-mark scenario question: mark scheme

21 marks · Cambridge 0478 syllabus: Paper 2 scenario question (section 3, assessment details), 7.4 (standard methods: totalling, counting, finding maximum), 7.5(a) (validation checks), 7.8 (identify and correct errors), 7.9 (write algorithms using pseudocode or program code), 8.1.3 (input and output), 8.1.4 (selection, iteration, totalling, counting), 8.1.8 (maintainable programs: identifiers and comments), 8.2.1–8.2.3 (1D and 2D arrays, reading and writing with iteration)

How to use. Questions 1 and 2: one mark per bullet point up to the maximum shown. **/** separates alternative words; **//** separates alternative answers. Words in (brackets) are not needed for the mark. Question 3 is marked as a whole using the two level grids below (best fit), then the AO2 and AO3 marks are added. Ignore minor syntax slips throughout. Example answers are one possible response, not the only one.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 8.2.1, 8.2.2	The number of bikes hired from the third station // from the station in `StationName[3]` … on day 5	**1**	Both the station and the day are needed. Do not accept station 5 on day 3. Do not accept "the total for station 3" or "the name of station 3".
1(b) 0478 7.4	R2: **Totalling** R3: **Counting** R4: **Finding the maximum**	**3**	One mark per correct row. A row with more than one tick scores 0.
2 0478 7.8, 7.5(a), 8.1.3, 8.1.8	One mark for each problem identified **with** a matching correction. Max 2. Any two from: • No prompt / message before the input (line 03 // line 07), so the user is not told what to enter … **correction:** add an `OUTPUT` before each `INPUT` asking for the number of hires, e.g. `OUTPUT "Enter hires for ", StationName[Station], " on day ", Day` • The validation uses an IF, so the value is only checked once // a second invalid value (e.g. 301 after −5) is accepted and stored … **correction:** use a condition-controlled loop (REPEAT … UNTIL // WHILE … DO … ENDWHILE) so the value is input again until it is from 0 to 300 • The error message on line 06 is not suitable // does not say what is wrong or what the valid range is … **correction:** output a meaningful message, e.g. `"Invalid: enter a number from 0 to 300"` • There are no comments, so the code is not explained … **correction:** add comments (`//`) to explain the loops and the validation	**2**	Max 2. The problem and its correction must match; a problem with no correction scores 0. Credit any other valid problem and correction. Do not credit: saying the condition on line 04 is wrong (it is correct); saying the loop bounds are wrong; saying the arrays must be declared (the question says they need not be); "it should use a different array".

Question 3 (15 marks): requirements, data structures and level grids

QUESTION	ANSWER	MARKS	GUIDANCE
3 0478 Paper 2 scenario question; 7.4, 7.5(a), 7.9, 8.1.3, 8.1.4, 8.1.8, 8.2.1–8.2.3	**Requirements (techniques)** **R1** Input the number of hires for each station for each day, with prompts, and store it in `Hires[]` (nested iteration, input with prompts, storing in a 2D array); validate each value is from 0 to 300 inclusive and require re-entry until valid (range check inside a condition-controlled loop) **R2** Total the hires for each station and store each total in `StationTotal[]` (nested iteration, totalling, storing in a 1D array); output each station's name with its total (output with messages) **R3** Count the stations whose total is less than 150 (iteration, selection, counting); output the count with a message **R4** Find the highest weekly total (iteration, selection, finding the maximum); output the highest total and the name of every station with that total (iteration and selection over the totals, output with messages) **Data structures required** (names must be used as given) 1D array `StationName[1:12]` (station names, already stored) 2D array `Hires[1:12, 1:7]` (hires for each station on each day) 1D array `StationTotal[1:12]` (weekly total for each station) Variables chosen by the candidate, e.g. loop counters, the value input, the count, the highest total	**15**	Pseudocode, Python, Visual Basic or Java are all accepted. Any other programming language scores 0. Syntax is not assessed: mark the logic. Built-in functions of the chosen language (e.g. `max()`, `sum()` in Python) may be used to meet a requirement. Arrays indexed from 0 are accepted when used consistently. Ignore any declarations. The given array names must be used; do not penalise a different case (e.g. `stationtotal`). Common ways to lose marks: validating with a single IF (checks once only); outputting only one station when several share the highest total; resetting a total or count inside the wrong loop; inputs with no prompts; no comments; using names other than the given arrays.

AO2: apply knowledge and understanding (max 9). Mark by best fit.

Level	Description
7–9	All the requirements R1 to R4 are covered by appropriate techniques; the data structures store all the data that is needed.
4–6	More than one appropriate technique is applied to the scenario; some appropriate data structures are used to store some of the data.
1–3	At least one appropriate technique is used; some data is stored, but not always in an appropriate way.
0	No creditable response.

AO3: provide solutions to problems (max 6). Mark by best fit.

Level	Description
5–6	The program is commented throughout and uses meaningful identifiers. It is in a logical order, is accurate and meets all the requirements R1 to R4. Minor syntax errors are ignored.
3–4	The program has some relevant comments and most identifiers are sensible. It contains some errors but performs most of the requirements.
1–2	The program has no relevant comments, but some identifiers are sensible. It is illogical or inaccurate in many places, but attempts at least one requirement.
0	No creditable response.

Total for Question 3 = AO2 mark (max 9) + AO3 mark (max 6) = **15**.

Question 3: example 15-mark answer (Cambridge pseudocode)

One possible answer, sectioned by requirement. Other logically correct answers, and answers in Python, Visual Basic or Java, gain full marks.

```
// R1: input and validate the hires for every station for every day
FOR Station ← 1 TO 12
    FOR Day ← 1 TO 7
        // prompt names the station and the day so the user knows what to enter
        OUTPUT "Enter the number of bikes hired from ", StationName[Station],
          " on day ", Day, " (0 to 300)"
        INPUT NumberHired
        // range check: keep asking until the number is from 0 to 300
        WHILE NumberHired < 0 OR NumberHired > 300 DO
            OUTPUT "Invalid: the number must be from 0 to 300. Enter it again"
            INPUT NumberHired
        ENDWHILE
        // store the valid number in the 2D array
        Hires[Station, Day] ← NumberHired
    NEXT Day
NEXT Station
```

```
// R2: total the hires for each station and output each total
FOR Station ← 1 TO 12
    // start each station's total at zero
    StationTotal[Station] ← 0
    FOR Day ← 1 TO 7
        StationTotal[Station] ← StationTotal[Station] + Hires[Station, Day]
    NEXT Day
    OUTPUT StationName[Station], " total hires this week: ", StationTotal[Station]
NEXT Station
```

```
// R3: count the stations with fewer than 150 hires in the week
QuietCount ← 0
FOR Station ← 1 TO 12
    IF StationTotal[Station] < 150
      THEN
        QuietCount ← QuietCount + 1
    ENDIF
NEXT Station
OUTPUT "Stations with fewer than 150 hires this week: ", QuietCount
```

```
// R4: find the highest weekly total
HighestTotal ← StationTotal[1]
FOR Station ← 2 TO 12
    IF StationTotal[Station] > HighestTotal
      THEN
        HighestTotal ← StationTotal[Station]
    ENDIF
NEXT Station
OUTPUT "Highest weekly total: ", HighestTotal
// output every station with the highest total, in case more than one has it
OUTPUT "Station(s) with this total:"
FOR Station ← 1 TO 12
    IF StationTotal[Station] = HighestTotal
      THEN
        OUTPUT StationName[Station]
    ENDIF
NEXT Station
```

Total: 21 marks