MODULE K ALGORITHM DESIGN · K4

The program development life cycle: mark scheme

21 marks · Cambridge 0478 syllabus (exams 2026-2028) 7.1 (understand the program development life cycle, limited to analysis, design, coding and testing) and 7.2(c) (methods to design a solution: structure diagrams, flowcharts, pseudocode)

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. The syllabus limits the life cycle to four stages: do not credit other stage names (e.g. maintenance, evaluation, implementation).

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) 0478 7.1	One mark for each correct stage in the correct position: Stage 2: Design Stage 3: Coding Stage 4: Testing	**3**	Accept "programming" // "writing (the) code" for Coding. Do not credit "implementation". A correct stage in the wrong box scores 0 for that box.
2(a) 0478 7.1	One mark for each correct row: Asking the centre manager what the program must do: Analysis Writing the program code for the cost module: Coding Drawing a structure diagram: Design Running the finished program with a group size of 9: Testing Deciding which details the program does not need: Analysis	**5**	A row with more than one box ticked scores 0. Row 5 is abstraction, which the syllabus places in analysis.
3(a) 0478 7.1	Any two from: Book a session of 1, 2 or 3 hours // input / store the session length Accept / check / validate a group size of 1 to 8 climbers Check that every climber has completed a safety induction (before booking them in) Output a booking reference Output the total cost (of the session)	**2**	max 2. The last note gives two requirements: credit the booking reference and the total cost separately. Do not credit any of the last three notes (walls, café, opening year). Credit any other valid requirement drawn from the notes.
3(b) 0478 7.1	Any two from: The walls were repainted // the colour of the walls The café (sells hot drinks) The year the centre opened // 2015	**2**	max 2. Do not credit any requirement listed in 3(a): those are needed by the program.

QUESTION	ANSWER	MARKS	GUIDANCE
4(a) 0478 7.1, 7.2(c)	One mark for each empty box containing a different valid sub-problem, e.g.: Enter / input the booking details // the session length // the number of climbers Check / validate the group size (1 to 8) // the session length (1, 2 or 3 hours) Calculate / work out the total cost Create / generate the booking reference Output the booking reference // the total cost // the booking details Store / save the booking	**2**	Credit any other valid sub-problem of the booking program. Do not credit: a repeat of the given box; the same sub-problem in both boxes; the whole problem ("book a session"); an item that is not part of the program (e.g. the walls or the café).
4(b) 0478 7.1, 7.2(c)	Flowchart(s) Pseudocode	**2**	Do not credit "structure diagram" (already used), "program code" or "trace table".
5(a) 0478 7.1	Any two from: Each module / part of the program is tested as (soon as) it is written // before the next module is written Modules are tested separately / individually (e.g. the cost module on its own) Any errors found are corrected // the code is changed ... and the module is tested again // testing is repeated until the module works (as expected)	**2**	max 2. Do not credit "testing the program" or "checking it works" alone: the answer must show testing is repeated / carried out during the writing of each part.
6(a) 0478 7.1 (7.6)	Any three from: Choose / create test data (for the program) // write a test plan ... using normal, abnormal, extreme and boundary data // a correct example in context (e.g. a group size of 4 as normal data; 1 or 8 as extreme data; 1 and 0 // 8 and 9 as boundary data (1 / 8 accepted, 0 / 9 rejected); 12 or "ten" as abnormal data) Work out the expected result / outcome for each test Run the (complete) program with the test data Compare the actual results with the expected results // record the results If a result is not as expected, find and correct the error (and test again) Check that the program meets (all) the requirements (from the analysis stage)	**3**	max 3. Credit an example only if everything the candidate gives is correct: any type of test data named must match the value, and any expected outcome must be correct (a group size of 1 to 8 is accepted; 0, 9 or a non-integer is rejected). Do not credit an example with the wrong type of test data (e.g. "9 as normal data"). Credit any other valid test-data example from the scenario (e.g. a session length). Do not credit "check it works" or "look for errors" alone.

Total: 21 marks