

Task 6 Marking Key

Task 6 Study Session Helper Program - Unit 4 - 20%

Task 6 Marking Key

Study Session Helper Program

This marking key explains how the 50 marks are awarded for Task 6. Award marks for the specific evidence listed in each criterion and use the marking notes to apply the key consistently.

Do not award marks for visual polish alone. Award marks for evidence that the student applied the Software Development Cycle, built a working coded Study Session Helper Program and documented the programming logic, testing and evaluation.

Required Project Evidence

The student should produce:

- a working coded Study Session Helper Program
- evidence document showing each Software Development Cycle stage
- source code, project file or product link
- prototype, wireframe or interface planning evidence
- input-process-output and variable/data planning evidence
- pseudocode and/or flow chart
- trace table
- testing log with expected and actual results
- validation and iteration evidence
- evaluation against requirements and success criteria
- acknowledgement of external help, assets, tutorials, examples, code snippets or AI support

Marking Allocation

Criterion	Marks
1. Problem definition, user requirements and scope	6
2. Product design and SDC planning	8
3. Algorithm design	8
4. Program development	14
5. Testing, debugging, validation and iteration	8
6. Evaluation, documentation and submission quality	6
Total	50

Criterion 1: Problem Definition, User Requirements And Scope (6 marks)

Required student evidence	Marking guidance	Marks
Identify the target user as a Year 12 student preparing for assessment or a clearly related Year 12 study user.	Award for a specific, appropriate target user.	1
State a clear study-session problem or need linked to the supplied stimulus.	Award for a problem/need connected to assessment preparation, revision, study focus, practice or next-step advice.	1
Describe at least four functional requirements linked to the Study Session Helper Program.	Award 1 mark for two or three relevant functional requirements; award 2 marks for at least four relevant functional requirements.	2
Identify relevant constraints, such as time, device, programming environment, user ability, available data or classroom conditions.	Award for realistic constraints that affect the product scope or development.	1
Write testable success criteria that can be used in evaluation.	Award for success criteria that can be checked, tested or judged in the final evaluation.	1

Marking notes:

- Full marks require the problem and requirements to remain linked to the Study Session Helper Program.
- Requirements must be testable. Vague statements such as “make it good” or “easy to use” need detail to earn full credit.

Criterion 2: Product Design And SDC Planning (8 marks)

Required student evidence	Marking guidance	Marks
Describe how the Software Development Cycle stages will be used for this project.	Award for applying the SDC to this project, not only listing the stages.	1
Provide a user flow or step-by-step interaction plan for the study session.	Award for a clear sequence of user actions and program responses.	1
Provide a wireframe, prototype, screen sketch or text-interface plan showing the main user interaction.	Award 1 mark for partial or limited prototype/interface evidence; award 2 marks for clear evidence showing the main interaction.	2
Provide an input-process-output plan connected to the required program features.	Award for IPO evidence connected to the Study Session Helper requirements.	1
Provide a data or variable plan with suitable names and intended data types.	Award for variable/data planning that is relevant to the final product.	1
Explain how user needs influenced the interface or interaction design.	Award for a clear link between the intended user and design decisions.	1
Keep the planned Study Session Helper Program scope realistic for the assessment window.	Award for a product scope that is achievable and still meets the required features.	1

Marking notes:

- Do not require Figma specifically. Paper, Keynote, Figma or text-interface planning can earn marks if the design thinking is clear.
- Prototype evidence should match the final product closely enough to show planning was useful.

Criterion 3: Algorithm Design (8 marks)

Required student evidence	Marking guidance	Marks
Provide pseudocode and/or a flow chart that represents the main Study Session Helper Program logic.	Award 1 mark for partial algorithm evidence; award 2 marks for clear evidence of the main logic.	2
Show sequence clearly.	Award for a logical order of steps.	1
Show at least one meaningful selection structure that affects advice, feedback, scoring or output.	Award for a decision that changes the program outcome.	1
Show at least one meaningful iteration structure for repeated questions, menu use, topic entry or attempts.	Award for repeated logic that suits the task.	1
Include validation or error-handling logic in the algorithm design.	Award for checking or responding to invalid/unexpected input.	1
Provide a trace table for a meaningful part of the algorithm.	Award for a trace table that tracks variable values or decision outcomes.	1
Make sure the algorithm evidence is clear and consistent with the final code.	Award for algorithm evidence that matches the submitted product.	1

Marking notes:

- A trivial selection, such as checking only whether the user pressed start, should not receive full selection credit unless it affects the product outcome.
- A trace table must track variable values or decision outcomes. A simple screenshot of output is not a trace table.

Criterion 4: Program Development (14 marks)

Required student evidence	Marking guidance	Marks
Runs correctly or has authenticated evidence of running.	Award 1 mark for partially running or authenticated partial evidence; award 2 marks for a working product.	2
Is recognisably a Study Session Helper Program and addresses the supplied stimulus.	Award 1 mark for a weak or partial link to the stimulus; award 2 marks for a clear Study Session Helper product.	2
Uses input, processing and output clearly.	Award 1 mark for partial IPO; award 2 marks where input, processing and output are all clear.	2
Uses meaningful variable names and suitable data types.	Award 1 mark for limited use; award 2 marks for meaningful names and suitable data/data type choices.	2
Uses sequence and readable program structure.	Award for clear program order and organisation.	1
Uses at least one meaningful selection structure in working code.	Award 1 mark for limited selection; award 2 marks for meaningful selection that changes advice, feedback, scoring or output.	2
Uses at least one meaningful iteration structure in working code.	Award 1 mark for limited repetition; award 2 marks for meaningful iteration suited to questions, menus, topics or attempts.	2
Includes validation, error handling or user-friendly handling of unexpected input.	Award for checking or handling at least one important input or unexpected action.	1

Marking notes:

- If the product does not run, award only marks that can be verified from source code, screenshots, checkpoint evidence or teacher-authenticated demonstration.
- A modest but complete program can score highly if it clearly shows the required programming concepts.
- Do not award full marks for copied or heavily scaffolded code unless the student has adapted it and can explain it.

Criterion 5: Testing, Debugging, Validation And Iteration (8 marks)

Required student evidence	Marking guidance	Marks
Include testing with normal data.	Award for a relevant successful/typical test.	1
Include testing with boundary data where relevant, such as minimum/maximum time, lowest/highest confidence rating or final quiz question.	Award for a relevant boundary or edge-case test.	1
Include testing with invalid or unexpected data where relevant, such as blank topic, impossible time or rating outside the accepted range.	Award for a relevant invalid/unexpected input test.	1
Record expected and actual results.	Award 1 mark for limited expected/actual result evidence; award 2 marks where results are recorded clearly across the testing evidence.	2
Identify at least one error, issue or design problem and explain how it was addressed.	Award for specific debugging or problem-solving evidence.	1
Include validation evidence that checks the product against user needs, success criteria or user feedback.	Award for checking suitability, not just technical correctness.	1
Show at least one improvement or justified change.	Award for documented iteration based on testing, validation or reflection.	1

Marking notes:

- Do not award full testing marks for a list of tests with no actual results.
- If boundary data is not relevant to the product, award the boundary mark for a clearly justified alternative edge-case test.

Criterion 6: Evaluation, Documentation And Submission Quality (6 marks)

Required student evidence	Marking guidance	Marks
Evaluate the final product against success criteria or requirements.	Award 1 mark for limited evaluation; award 2 marks for clear judgement against success criteria or requirements.	2
Identify specific strengths and limitations of the final product.	Award for specific strengths and limitations, not generic comments.	1
Recommend realistic future improvements.	Award for improvements that suit the product and remaining limitations.	1
Organise and name evidence clearly so it can be marked.	Award for a complete, accessible and clearly organised submission.	1
Acknowledge external help, assets, tutorials, examples, code snippets or AI support where used.	Award for clear acknowledgements, or a statement that none were used.	1

Marking notes:

- Evaluation must make a judgement. A description of what the product does is not enough for full marks.
- If there is no external help, the student should state that no external help was used.

Non-Award And Authentication Guidance

- Do not award marks for evidence that cannot be opened, viewed or verified unless checkpoint evidence confirms it.
- If final code is missing, use authenticated class evidence only where available and record this on the marking sheet.
- If source code is copied without acknowledgement or cannot be explained, apply the school assessment policy and record the issue.
- If a student changes the product purpose away from the required Study Session Helper Program without approval, mark the evidence that is present but do not assume missing planning evidence.

Prepared for Task 6: Study Session Helper Program.