THE KING'S COLLEGE - YEAR 12 COMPUTER SCIENCE GENERAL

Task 6 Study Session Helper Brief

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

Task 6 Study Session Helper Brief

Study Session Helper Program

Supplied Stimulus

Year 12 students often prepare for multiple assessments in the same term. Some students know they need to revise, but they do not always know where to start, how long to spend or which topic needs attention first.

The King's College wants a small digital tool that helps a Year 12 student run a short study session. The program should collect study session information, provide practice prompts or questions, track progress and give the student a useful summary or recommendation.

Product Goal

Create a working coded **Study Session Helper Program** that:

- is designed for a Year 12 student preparing for assessment
- asks the user for study session information
- includes at least three practice questions, flashcards or prompts
- tracks score, attempts, progress or priority using variables
- gives feedback or recommendations using selection
- repeats questions, topics or menu choices using iteration
- checks at least one input using validation or error handling
- produces a final study summary or next-step recommendation
- is designed, tested, improved and evaluated using the Software Development Cycle

Minimum Program Requirements

Your program must include:

Requirement	What this means
Input	The user enters study topic, available study time and a confidence or priority rating.
Processing	The program calculates, checks, scores, categorises or recommends based on user input.
Output	The program displays feedback, score, progress, a study plan or next-step advice.
Variables	The program stores values such as topic, time, rating, score, attempts or recommendation.
Suitable data types	Text, numbers and Boolean/true-false values are used where suitable for the chosen environment.
Sequence	The program follows a logical order from start, input, processing, output and summary.
Selection	A decision changes the output, such as different advice for low, medium or high confidence.
Iteration	A loop or repeated action allows multiple questions, repeated menu choices or repeated attempts.
Validation	The program checks one important input, such as blank topic, impossible time or rating outside the accepted range.
Interface	The user interaction is clear, labelled and suitable for a Year 12 student.
Documentation	Internal comments and external evidence explain the logic and design decisions.
Testing	Normal, boundary and invalid test data are used where relevant.
Evaluation	The final product is judged against requirements and success criteria.

Required Stages

Stage	SDC link	What you produce
Define	State the problem	identify the target user, state the study-session problem, describe at least four functional requirements, identify constraints and write testable success criteria
Plan and design	Plan and design	describe how the SDC will be used, provide a user flow, wireframe/interface plan, input-process-output plan, data/variable plan and user-need design explanation
Algorithm design	Plan and design	provide pseudocode and/or a flow chart showing sequence, selection, iteration, validation/error handling and a trace table
Build	Develop the solution	submit a running coded Study Session Helper Program with input, processing, output, variables, suitable data choices, sequence, selection, iteration and validation/ error handling
Test and validate	Test the solution	include normal, boundary and invalid tests, expected and actual results, debugging evidence, validation evidence and at least one improvement
Iterate and evaluate	Evaluate the solution	evaluate against success criteria or requirements, identify strengths and limitations, recommend improvements, organise evidence and acknowledge external help where used

Example Study Session Flow

Your product does not need to copy this exact flow, but it must meet the required features.

1. Display a welcome screen or menu.
2. Ask for the student's name or session label.
3. Ask for the study topic.
4. Ask how many minutes the student has available.
5. Ask for a confidence rating, such as 1 to 5.
6. Validate the rating and time input.
7. Ask at least three practice questions or show at least three flashcards/prompts.
8. Track score, attempts, completed prompts or confidence changes.
9. Use selection to give different feedback or recommendations.
10. Display a final summary with the student's focus area and next step.

Challenge Features

Add challenge features only after the core program works.

Possible challenge features:

- menu system
- topic categories
- random question order
- difficulty levels
- scoring or progress percentage
- retry logic
- summary report

- reusable functions or modular sections
- saved data, if approved by your teacher
- improved visual layout

Scope Rules

Keep the product small enough to finish. Your marks come from the quality of your process, code and evidence, not from making the most ambitious product.

Avoid:

- building a full revision website with many pages
- creating a polished interface with little working logic
- adding many questions before selection, iteration and validation work
- using code you cannot explain
- submitting a Figma prototype instead of a working coded product
- leaving testing or evaluation until the final lesson

Teacher Approval

Your teacher must approve:

- programming environment
- planned interface type
- practice question or prompt set
- minimum feature list
- planned evidence

Do not begin the final build until your Study Session Helper Program plan has been approved.
