Reading criteria and interpreting unseen stimulus

Unseen does not mean unknowable

Unseen stimulus gives you a new context, not a new subject. Your job is to connect supplied evidence with familiar Digital Solutions concepts.

> **First principle:** build your answer from the stimulus and the question, then apply subject knowledge. Do not replace supplied evidence with a memorised response about a similar system.

Expect useful information to be distributed across a scenario, criteria list, diagram, table, interface, pseudocode or data sample. A relationship may only become clear when two sources are combined.

Fact or assumption?

A locker app uses a six-digit code and depends on one network connection. **Determine** which statement is directly supported.

1. The code is encrypted with AES.
2. A network failure could prevent code verification.
3. Every user has a weak password.

Check your answer

Statement 2. The single network dependency supports an availability risk. The encryption method and user passwords were not described.

Classify what you read

Information type	What it tells you	Example
Fact	A stated feature, event or condition	The app refreshes every 20 minutes.
Requirement	What the solution must do or how well it must perform	Display a warning within five minutes.
Criterion	Observable evidence used to judge success	A test warning appears within 300 seconds.
Constraint	A limit on possible solutions	Existing school tablets must be used.
Test evidence	What occurred when a component was tested	The warning appeared after 18 minutes.

A test result only becomes meaningful when compared with a criterion. Eighteen minutes is neither good nor bad until the required time is known.

Sort a short stimulus

A school has six weeks to create a web app using its existing server. The app must return matching resources within two seconds. A test with 1000 records takes 3.4 seconds.

Recognise one time constraint, one technical constraint, one criterion and one test result.

Check your answer

- Time constraint: six weeks.
- Technical constraint: use the existing server.
- Criterion: matching resources returned within two seconds.
- Test result: 3.4 seconds with 1000 records.

Turn criteria into observable checks

A useful criterion states the expected behaviour, relevant condition and observable result. Vague goals such as “easy to use” are difficult to test.

Vague statement	Observable criterion
Search is fast.	For 5000 records, a valid search displays matching results within two seconds.
Errors are clear.	Each invalid required field displays a specific message beside that field without deleting valid input.
Data is protected.	A student account cannot view or change another student's private record.

Make it testable

The requirement says, “The interface must clearly show offline records.” **Interpret** this as one observable criterion.

View a sample solution

When the device is offline, each unsynchronised record displays an icon and the text “Not synced”, and remains available for retry when connectivity returns.

Read across sources

Use a reference map when evidence is spread across formats.

1. Identify the named component in the question.
2. Find it in each relevant source.
3. Trace identifiers, labels and flows that connect the sources.
4. Record only facts that help answer the question.

For example, a mock-up may show a *Save report* button, pseudocode may show the validation it triggers, and a data dictionary may reveal which fields are required. Together they explain the complete feature.

> **Match exact identifiers:** an ID may connect a user-interface selection, JSON object, database record and DFD flow even when their layouts look unrelated.

Combine three sources

A mock-up contains a required *room* list and *Submit* button. A data specification states `room_id: INTEGER, required`. Pseudocode checks that `room_id` exists before saving.

Explain how the three components work together.

Check your answer

The user selects a room through the list, which supplies the required integer `room_id`. When Submit is activated, the programmed component checks that the identifier exists before writing the record. The UI, data constraint and validation therefore enforce the same requirement at different parts of the solution.

Annotate with a purpose

On paper, use a small set of consistent marks rather than highlighting everything.

- Box cognitive verbs and requested quantities.
- Underline criteria and constraints.
- Circle named entities, processes, fields or functions.
- Use short margin labels such as *C* for confidentiality or *EV* for evaluation evidence.
- Draw a line between matching identifiers across sources.

Annotations should reduce rereading. If every sentence receives the same mark, no information has been prioritised.

Build an evidence table

A useful scratch table can keep reasoning aligned:

Demand	Evidence	Meaning	Conclusion
Evaluate availability	Only server; no offline mode	One failure blocks all access	Weak availability

Resolve apparent contradictions

Two pieces of stimulus may describe different states or conditions rather than truly conflict.

- A requirement describes the intended result; a test result describes actual behaviour.
- A DFD shows logical movement; a network diagram shows physical connections.
- A mock-up shows a proposed interface; a screenshot may show the current interface.
- A general rule may be narrowed by a scenario-specific constraint.

State the distinction explicitly. For example: “The criterion requires five minutes, but the observed 18-minute result shows the current implementation does not meet it.”

Independent practice: library collection lockers

Read this unfamiliar system description.

- Members reserve books in a mobile app and collect them after hours.
- The app displays a six-digit collection code.
- A locker terminal requires a continuous network connection to verify the code.
- The criterion states that an authorised member can open the correct locker within 20 seconds.
- Test A: a valid code opens the correct locker in 12 seconds.
- Test B: during a network interruption, the terminal displays a spinning icon for 90 seconds and does not open.
- The terminal keeps the previous member's name and book title visible until another code is entered.

- Staff manually copy locker numbers from a spreadsheet into the reservation system.

On paper:

1. **Recognise** one functional requirement, one criterion and two pieces of test evidence.
2. **Evaluate** the system against the 20-second criterion.
3. **Determine** one confidentiality, one integrity and one availability risk. Support each with different evidence.
4. **Explain** why the two test results do not contradict each other.

View a sample solution

1. Functional requirement: verify a collection code and open the correct locker. Criterion: an authorised member opens it within 20 seconds. Evidence: 12 seconds with a valid code and no opening after 90 seconds during interruption.
2. The system meets the criterion under the connected condition in Test A but fails during the interruption in Test B. It therefore does not reliably meet the criterion across the described operating context.
3. **Confidentiality:** the previous member's name and title remain visible to the next person. **Integrity:** manually copying locker numbers may create a wrong reservation-to-locker link. **Availability:** the continuous network dependency prevents authorised collection during an interruption.
4. The tests use different network conditions. Test A demonstrates successful behaviour while connected; Test B reveals failure when the required connection is unavailable.

Revision #5
Created 2026-10-06 00:40:24 UTC by Mr Napper
Updated 2026-10-06 01:32:05 UTC by Mr Napper