



Certified Low-Code Developer (CLCD)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working low-code developer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: Low-Code Foundations

DATA

Emma Wilson matches each low-code building block to its role.

Building block	Role
Visual designer	(to choose)
Data table	(to choose)
Connector	(to choose)
Workflow	(to choose)

Table 1.1 Building blocks and their roles.

Which set of roles is correct?

- ☐ A Stores records, builds the screens, automates steps and links to outside services, so the connector is the block that stores records.
- ☐ B Links to outside services, automates steps, builds the screens and stores records, so the workflow is the block that builds screens.
- ☐ C Builds the screens, stores records, links to outside services and automates steps, matching each block to the role it truly serves.
- ☐ D Automates steps, links to outside services, stores records and builds the screens, so the visual designer is what automates the steps.

Correct answer: C

The visual designer builds the screens, the data table stores records, the connector links to outside services, and the workflow automates steps. Only option C pairs each block with the role it actually serves.

Question 2

Domain 1: Low-Code Foundations

CONFIG

James Carter reviews a plan that hand-codes a simple intake form from scratch.

```
requirement : simple intake form
team skill  : business analysts
timeline    : two weeks
current plan : custom code build
```

Listing 2.1 The current plan.

What is the soundest change?

- ☐ A Hand-code the whole app in a full programming language, since custom code is always faster to deliver than any low-code build.
- ☐ B Build the form on a low-code platform, so the business analysts can assemble it visually and meet the two-week timeline.
- ☐ C Cancel the project outright, because a simple intake form can never be delivered by a team of business analysts in any timeframe.
- ☐ D Raise the budget and hire more developers, on the assumption that adding more coders alone will deliver a simple form faster.

Correct answer: B

Low-code platforms let people with business skills build a simple form visually, which fits both the team and the two-week timeline. Custom code would be slower here, cancelling wastes a small task, and more coders does not suit a low-code job.

Question 3

Domain 2: Building Apps

DATA

Olivia Davis maps each app-building task to the right component.

Task	Component
Show a list of records	(to choose)
Collect user input	(to choose)
Trigger a save action	(to choose)
Move between screens	(to choose)

Table 3.1 Tasks and components.

Which set of components is correct?

- ☐ A A list view for records, a form for input, a button for the save action, and navigation to move between screens.

- B** A button for records, navigation for input, a form for the save action, and a list view to move between the app screens.
- C** A form for records, a button for input, navigation for the save action, and a list view to move between the app screens.
- D** Navigation for records, a list view for input, a button to move screens, and a form to trigger the save action reliably.

Correct answer: A

A list view shows records, a form collects input, a button triggers the save action, and navigation moves between screens. Only option A pairs each task with the component built for it.

Question 4

Domain 3: Data and Logic

CONFIG

Liam Anderson lists the steps to add a calculated total, shown out of order.

- A. The total appears in the list view
- B. Add a number field for quantity
- C. Add a calculated field for total
- D. Set the formula to price times quantity
- E. Add a number field for price

Listing 4.1 Steps to order.

What is the correct order of these steps?

- A** C, then D, then E, then B, then A, creating the calculated field and its formula before either number field it reads exists.
- B** D, then C, then E, then B, then A, writing the formula before the calculated field that holds it or the number fields it reads.
- C** A, then E, then B, then C, then D, showing the total before the fields and the formula that produce the value have been created.
- D** E, then B, then C, then D, then A, adding both number fields, the calculated field, its formula, then showing the total.

Correct answer: D

You add the price and quantity number fields, add the calculated field, set its formula to price times quantity, and then the total appears in the list view. The other orders build the formula or show the total before the fields it depends on exist.

Question 5

Domain 4: Automation and Workflows

DATA

Ava Thompson matches each workflow part to its job.

Part	Job
Trigger	(to choose)
Condition	(to choose)
Action	(to choose)
Connector	(to choose)

Table 5.1 Workflow parts and jobs.

Which set of jobs is correct?

- ☐ A Does the work, starts the flow, reaches an outside service and decides the branch, so the trigger is what does the actual work.
- ☐ B Starts the flow, decides the branch, does the work and reaches an outside service, matching each part to the job it performs.
- ☐ C Decides the branch, reaches an outside service, starts the flow and does the work, so the condition is what starts the flow off.
- ☐ D Reaches an outside service, does the work, decides the branch and starts the flow, so the connector is what starts the flow off.

Correct answer: B

The trigger starts the flow, the condition decides which branch to take, the action does the work, and the connector reaches an outside service. Only option B pairs each part with the job it performs.

Question 6

Domain 5: Integration and Extensibility

DATA

David Brown matches each integration need to the right approach.

Need	Approach
Read data from another app	(to choose)
React to an event elsewhere	(to choose)
Add logic beyond the platform	(to choose)
Reuse one call in many apps	(to choose)

Table 6.1 Integration needs and approaches.

Which set of approaches is correct?

- ☐ A A connector to read data, a webhook to react to events, a custom step for extra logic, and a shared component to reuse calls.
- ☐ B A webhook to read data, a shared component to react to events, a connector for extra logic, and a custom step to reuse a call.
- ☐ C A custom step to read data, a connector to react to events, a shared component for extra logic, and a webhook to reuse a call.

- D** A shared component to read data, a custom step to react to events, a webhook for extra logic, and a connector to reuse a call.

Correct answer: A

A connector reads data from another app, a webhook reacts to an event elsewhere, a custom code step adds logic beyond the platform, and a shared component lets one call be reused across apps. Only option A pairs each need with the fitting approach.

Question 7 Domain 2: Building Apps

An app screen loads slowly because it pulls every record before showing the first page. What is the soundest fix?

- A** Leave the screen as it is, because loading every record up front is the only way to be sure the list is always complete for a user.
- B** Remove the list from the screen entirely, since a screen with no records to load at all will always open quickly for every user.
- C** Load records in small pages and filter to what the user needs, so the screen fetches far fewer records and opens quickly.
- D** Add more screens to the app, on the assumption that spreading records across screens alone makes each one load faster somehow.

Correct answer: C

Paging and filtering fetch only the records the user needs right now, so the screen loads a small set and opens quickly. Loading everything up front is the cause, removing the list defeats the screen, and extra screens do not reduce the data pulled.

Question 8 Domain 6: Governance, ALM and Security

A maker wants to share an app that reads sensitive HR data with the whole company. What should govern who can open it?

- A** Share the app with everyone by default, because an app that all staff can open easily is always the most useful to the business.
- B** Email the app link to the whole company, since sending the link widely is the simplest way to let the right people find the app.
- C** Turn off all security on the data source, so the app never has to check who is opening it or which records they may see.
- D** Grant access by role with least privilege, so only staff who need the HR data can open the app and its records stay protected.

Correct answer: D

Sensitive data calls for role-based access and least privilege, so only staff who need the HR data can open the app. Sharing with everyone, mailing the link widely, or removing security on the source would all expose data that must stay protected.

Question 9

Domain 4: Automation and Workflows

A scheduled workflow sometimes fails midway and leaves records half-updated, with no trace of what went wrong. What is required?

- ☐ A Run the workflow far more often, on the assumption that frequent runs alone will eventually correct any records left half-updated.
- ☐ B Add error handling and logging so a failed run is caught, recorded, and retried or rolled back instead of leaving bad data.
- ☐ C Delete the workflow whenever it fails, since removing it is the surest way to stop any half-updated records from appearing again.
- ☐ D Ignore the failures and fix records by hand, because manual cleanup after each failed run is the most reliable long-term plan.

Correct answer: B

Error handling and logging catch a failed run, record what happened, and either retry it or roll back the partial changes, so records do not stay half-updated. Running more often, deleting the flow, or manual cleanup all leave the underlying fault in place.

Question 10

Domain 6: Governance, ALM and Security

A team edits apps straight in the live production environment, and a bad change recently broke it for users. What practice fixes this?

- ☐ A Build and test in a separate environment, then promote a change to production only after it passes checks, keeping live stable.
- ☐ B Keep editing directly in production on every change, because fixing the live app at once is the fastest way to keep quality high.
- ☐ C Freeze the app forever after launch, since any later change to a working production app is certain to break it for the users again.
- ☐ D Let every maker publish to production freely, on the assumption that more people shipping changes alone raises overall quality.

Correct answer: A

Building and testing in a separate environment and promoting a change to production only after it passes checks keeps the live app stable. Editing production directly is the cause, freezing forever forgoes real fixes, and free publishing spreads the risk.