



Certified Intelligent Automation Specialist (CIAS)

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 intelligent automation specialist 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: Intelligent Automation Foundations

DATA

Emma Wilson matches each automation component to its job.

Component	Job
Attended bot	(to choose)
Unattended bot	(to choose)
Orchestrator	(to choose)
Work queue	(to choose)

Table 1.1 Components and their jobs.

Which set of jobs is correct?

- ☐ A Runs on a server on a schedule, runs beside a person, holds work items, and schedules bots, so a queue is what triggers the attended bot on a desktop.
- ☐ B Runs beside a person on demand, runs on a server on a schedule, schedules bots and holds credentials, and holds work items for bots to process in turn.
- ☐ C Schedules bots and holds credentials, holds the work items, runs beside a person, and runs on a server, so the orchestrator is really the attended bot.
- ☐ D Holds work items for bots to run, schedules the bots, runs on a server on a schedule, and runs beside a person, so the queue hands out the credentials.

Correct answer: B

An attended bot runs beside a person on demand, an unattended bot runs on a server on a schedule, the orchestrator schedules bots and holds credentials, and the work queue holds items for bots to process. Only option B pairs each component with its actual job.

Question 2

Domain 1: Intelligent Automation Foundations

CONFIG

James Carter reviews a request to automate a process, profiled below.

```
volume      : very low
exceptions  : very high
ui changes  : frequent
rules       : unclear
```

Listing 2.1 The current process profile.

What is the soundest decision?

- A** Automate the process exactly as it stands right now, because any manual process can be automated and the bot will simply handle each exception itself.
- B** Add several more bots to the process so the very high exception rate is shared out, since more bots always cancel out unclear rules and UI changes.
- C** Fix and stabilize the process first, then automate the stable parts, since a low-volume unstable process with many exceptions adds cost and rework.
- D** Raise the bot speed and run it far more often, on the assumption that running an unstable low-value process faster is what returns the most value here.

Correct answer: C

A low-volume process with very high exceptions, frequent UI changes and unclear rules is a poor candidate as it stands. The sound move is to fix and stabilize the process first, then automate the stable parts. Adding bots, running faster, or automating chaos raises cost and rework.

Question 3

Domain 2: Process Discovery and Assessment

DATA

Olivia Davis maps each discovery finding to the right action.

Finding	Action
High volume, stable rules, digital input	(to choose)
Needs judgement on most cases	(to choose)
Changes constantly, no fixed rules	(to choose)
Rare, one-off task	(to choose)

Table 3.1 Findings and actions.

Which set of actions is correct?

- A** Automate as a strong candidate, keep a human in the loop, hold until it is stable, and deprioritize as low return, matching each finding to its action.
- B** Deprioritize the task, automate it fully, keep it fully manual, and automate at once, so a rare one-off task is treated as the strongest candidate here.
- C** Keep it fully manual, deprioritize it, automate it fully, and hold for later, so a high-volume stable rules-based process is kept manual not automated.
- D** Hold for later, automate it fully, deprioritize it, and keep manual, so a process that changes constantly is automated before it has been stabilized.

Correct answer: A

A high-volume, stable, digital process is a strong automation candidate, judgement-heavy work needs a human in the loop, a constantly changing process should wait until stable, and a rare one-off task is low return. Only option A pairs each finding with the sound action.

Question 4

Domain 3: Building Automations

CONFIG

Liam Anderson lists the stages of processing one queue item, shown out of order.

- A. Mark the item complete and commit the result
- B. Get the next item from the work queue
- C. Do the work with try and catch on each step
- D. On error, log details and set retry or failed
- E. Open the target application and verify ready

Listing 4.1 Stages to order.

What is the correct order of these stages?

- A** B, then E, then C, then A, then D, marking the item complete before the work runs and logging any error only after the item is committed as done.
- B** E, then B, then A, then C, then D, opening the application before an item exists and marking the item complete before any work has actually been done.
- C** C, then D, then B, then E, then A, doing the work before an item is fetched or the application has been opened and verified ready for the bot to act.
- D** B, then E, then C, then D, then A, fetching the item, opening and verifying the app, working with try and catch, handling any error, then completing.

Correct answer: D

The flow fetches the item (B), opens and verifies the application (E), does the work with try and catch (C), logs and sets retry or failed on error (D), and marks the item complete on success (A). The other orders complete or work on items before they exist or before the app is ready.

Question 5

Domain 4: Integrating AI and Documents

DATA

Ava Thompson matches each extraction outcome to its routing decision.

Extraction outcome	Routing
Above the auto-post threshold	(to choose)
Between review and auto thresholds	(to choose)
Below the low-confidence floor	(to choose)
A required field is missing	(to choose)

Table 5.1 Outcomes and routing.

Which set of routings is correct?

- A** Send to review, post it straight through, reject it, and post it anyway, so a required field that is missing is posted with no human ever checking it.
- B** Post it straight through, send it to human review, reject or reprocess it, and route it to a human, matching each outcome to the correct routing decision.
- C** Reject everything by default, post it straight through, send it to review, and reprocess, so a high-confidence field above the threshold is rejected.
- D** Route it to a human, reject it, post it straight through, and send to review, so a low-confidence field below the floor is posted straight through as is.

Correct answer: B

A field above the auto-post threshold can post straight through, a mid-confidence field goes to human review, a field below the floor is rejected or reprocessed, and a missing required field is routed to a human. Only option B matches each outcome to a sound routing decision.

Question 6

Domain 5: Orchestration, Testing and Deployment

DATA

An automation is tested as shown below.

Observation	Result
Run against real sample documents	Never
Passed a single happy-path demo	Yes
Handles varied real exceptions	Low
Re-tested after a platform update	No

Table 6.1 How quality is checked today.

What is the core weakness, and what should be fixed?

- A** Quality rests on one demo; build a test set from real documents, run it before each release, and re-run it after every platform or model update made.
- B** Nothing is wrong here, because one happy-path demo that passed is enough to show the bot will handle the varied real exceptions it meets in production.
- C** The bot is simply too slow, so running it on faster infrastructure would raise its accuracy on varied real exceptions with no need to test beforehand.

- D** The process is far too long, so cutting steps from the automation alone would make it correct on the varied real exceptions where it now fails today.

Correct answer: A

Relying on a single happy-path demo with no test set from real documents means quality is judged by anecdote, which is why varied exceptions score low and a silent platform update goes unnoticed. The fix is a representative test set run before each release and re-run after every update.

Question 7 Domain 3: Building Automations

A bot reaches an item where a required approval is missing, a valid business situation and not a system fault. What is the soundest way to handle it?

- A** Retry the same item many times in a row at once, on the assumption that repeating a missing-approval item will eventually make the approval appear.
- B** Stop the whole automation and every other item, since one item without an approval means the entire queue can no longer be processed safely at all now.
- C** Mark it a business exception, skip it with a logged reason, and route it for human follow-up, while the bot carries on with the remaining queue items.
- D** Delete the item from the queue quietly, since an item that is missing an approval is not worth recording and can be removed without telling anyone.

Correct answer: C

A missing approval is a business exception, not a system fault, so the bot should log the reason, skip the item, route it for human follow-up, and continue the queue. Retrying blindly wastes runs, stopping everything blocks valid items, and deleting the item hides a real case.

Question 8 Domain 6: Governance, Security and Operations

A developer hard-codes a system password directly into the bot script so it can log in to the target application. What failed, and what is required?

- A** Nothing failed at all, because storing the password inside the bot script is the simplest way to log in and needs no vault or extra approval to stay safe.
- B** The bot platform was too small, and a more capable platform would keep the hard-coded password safe inside the script with no need for a managed vault.
- C** The script was simply too short, so adding more comments and logging around the password line is the complete fix and no credential store is needed here.
- D** Credentials must never be hard-coded; store them in a managed vault, grant the bot least-privilege access, and rotate and audit the secrets over time.

Correct answer: D

A hard-coded password can leak through source control and cannot be rotated cleanly. Secrets belong in a managed vault, with the bot given least-privilege access and the secrets rotated and audited. Platform size, comments, and logging do not control how credentials are stored.

Question 9

Domain 4: Integrating AI and Documents

An invoice bot posts every extracted total automatically, including totals the model flagged as low confidence, and wrong totals reach the ledger. What is required?

- ☐ A Nothing at all is required here, because a total the model extracted should always be posted automatically no matter how low its reported confidence is.
- ☐ B Add a confidence threshold and a human-in-the-loop step, so low-confidence totals go to a person for review before anything at all reaches the ledger.
- ☐ C Post every extracted total even faster, since the errors come from the bot running too slowly and a quicker run would keep wrong totals off the ledger.
- ☐ D Switch off extraction and key every invoice by hand once more, since routing only the uncertain totals to a person cannot be made to work reliably.

Correct answer: B

Posting low-confidence totals without review lets errors reach the ledger. A confidence threshold with a human-in-the-loop step routes uncertain totals to a person before posting, catching mistakes. Running faster does not raise accuracy, and re-keying everything discards the value of extraction.

Question 10

Domain 5: Orchestration, Testing and Deployment

A team edits the live production bot directly whenever it breaks, and its stability drifts up and down over time. What practice best fixes this?

- ☐ A Version the bot and promote changes through a test stage to production, releasing a change only when it passes tests without breaking other processes.
- ☐ B Keep editing the live bot on every break, because fixing production directly the moment it fails is the fastest route to a stable automation over time.
- ☐ C Freeze the bot for good right after launch, since any later change to a working automation is certain to make it less stable than it already is today.
- ☐ D Run the live bot far more often after each break, on the assumption that a higher run frequency alone will smooth out the drift in its stability soon.

Correct answer: A

Bots should be versioned and changes promoted through a test stage to production, so a change ships only when it passes tests without regressing other processes. Editing the live bot by anecdote causes the drift, freezing it forgoes fixes, and running more often does not improve stability.