Welcome to Cert007 - Your Ultimate IT Certification Partner

- Real Exam Questions
- Free Updates
- Expert Support
- Instant Access
- Money-Back Guarantee

Visit us at **https://www.cert007.com/** for more information

Exam : **Professional Agentic Architect**

Title : Professional Agentic Architect

Version : DEMO

1.A Dialogflow CX workflow uses Flow-Shipping-A. Its current page is Page-Address-A, which is not the flow start page. The session parameter address_verified is true, and the current turn has no matched intent. The architect wants the workflow to transition automatically to Page-Confirmation-A when address_verified is true, without requiring another user utterance. All pages named below belong to Flow-Shipping-A. No other routes or event handlers can trigger a transition in this fixture.
Which configuration provides the required transition in the current state?
A. Attach a condition-only route for address_verified = true to the current Page-Address-A, targeting Page-Confirmation-A.
B. Attach a route requiring the VerifyAddress intent and address_verified = true to Flow-Shipping-A, targeting Page-Confirmation-A.
C. Attach a condition-only route for address_verified = true to Flow-Shipping-A, targeting Page-Confirmation-A.
D. Attach a condition-only route for address_verified = true to the inactive Page-Review-A, targeting Page-Confirmation-A.
Answer: A
Explanation:
The deciding issue is handler scope at a non-start page. A condition-only route on the current page is in scope and can apply its transition when the condition is true. A flow-level condition-only route is limited to the flow start page, and a route on an inactive page is outside the current scope. Adding an intent requirement to a flow route makes its scope applicable away from the start page, but the current turn still needs a matching intent before that route can run.
A - Correct: The route belongs to the current page and its condition is true. It needs no intent match, so it can apply the transition to Page-Confirmation-A in the stated state.
B - Incorrect: The flow-level route has an intent requirement and therefore can be in scope away from the start page, but scope alone is insufficient. No intent matched in this turn, so its requirements are not satisfied.
C - Incorrect: A flow-level route without an intent requirement is in scope at the flow start page, not at the current non-start page. The true session parameter does not expand its scope.
D - Incorrect: A page route is in scope when its associated page is current. Page-Review-A is inactive, so its condition-only route cannot implement this transition now.

2.A Dialogflow CX architect is tracing one turn in Flow-Order-A. Page-Selection-A is current and is not the flow start page. The turn matched intent SelectItem, and the session parameter ready is true. The complete route fixture is shown below; within each scope the rows are in configured order.

Route	Scope	Requirement	Fulfillment	Transition target
Route-1	Current page	Intent SelectItem	Queue "Selected"	None
Route-2	Current page	Intent SelectItem	Queue "Repeated"	Page-Cancel-A
Route-3	Current page	Condition ready = true; no intent	Queue "Ready"	None
Route-4	Current page	Condition ready = true; no intent	Queue "Review"	Page-Review-A
Route-5	Current page	Condition ready = true; no intent	Queue "Extra"	None
Route-6	Active flow	Intent SelectItem	Queue "Flow selected"	None

All targets are pages in Flow-Order-A. There are no route groups, event handlers, webhooks, form-filling actions, or parameter changes. Target pages have no entry fulfillment or routes.
Which sequence of route fulfillments is called, and which page is current after this evaluation?
A. Route-1, Route-3, and Route-4; the current page becomes Page-Review-A.
B. Route-1 and Route-2; the current page becomes Page-Cancel-A.
C. Route-1, Route-6, Route-3, and Route-4; the current page becomes Page-Review-A.
D. Route-1, Route-3, Route-4, and Route-5; the current page becomes Page-Review-A.
Answer: A
Explanation:
Intent routes are evaluated before condition-only routes, with current-page intent routes preceding active-flow intent routes. Calling the first matching intent route consumes SelectItem. Its lack of a target allows evaluation to continue, but other routes requiring that consumed intent cannot run. The true condition is not consumed, so the first two matching condition-only routes can run. The transition on the second of those routes ends the current list's evaluation and activates Page-Review-A.
A - Correct: Route-1 consumes the matched intent but does not end evaluation because it has no target. Route-2 and Route-6 cannot reuse the consumed intent. Route-3 runs without consuming ready = true; Route-4 then runs and its transition ends evaluation before Route-5.
B - Incorrect: This sequence treats a target-free intent fulfillment as leaving the same intent available to a second page route. Route-1 already consumes SelectItem, so Route-2 cannot run or transition to Page-Cancel-A.
C - Incorrect: This sequence assumes that intent consumption applies only within page-level routes. The consumed match is unavailable to the active-flow intent route as well. The fixture includes no transition into another flow that could invoke the documented intent-propagation exception.
D - Incorrect: This sequence correctly allows multiple routes to use the true condition but ignores the transition boundary. Route-4 has a target, so evaluation of the list ends before Route-5 can add its fulfillment.

3.A Dialogflow CX workflow is currently filling the required postal_code parameter on Page-Address-A in Flow-Shipping-A. A user's reply does not satisfy that parameter, and sys.no-match-default is invoked. The complete no-match configuration is:

Handler scope	Event	Fulfillment	Target
postal_code parameter on the current page	sys.no-match-default	"Enter a valid postal code."	None
Current Page-Address-A	sys.no-match-default	"Please rephrase your request."	None
Active Flow-Shipping-A	sys.no-match-default	"Would you like general shipping help?"	None

There are no numbered no-match handlers or other handlers for this event. Later, while the same page remains current, the integration will invoke the custom event shipping.escalate. The architect wants that separate event to invoke a shipping-escalation fulfillment.
Which combination correctly predicts the invalid-input response and uses a supported scope for the custom event?
A. Expect "Enter a valid postal code."; add a shipping.escalate handler to the postal_code parameter.
B. Expect "Please rephrase your request."; add a shipping.escalate handler to Page-Address-A.
C. Expect "Would you like general shipping help?"; add a shipping.escalate handler to Flow-Shipping-A.

D. Expect "Enter a valid postal code."; add a shipping.escalate handler to Page-Address-A.
Answer: D
Explanation:
The invalid input is being handled during active form-parameter collection. The parameter-level reprompt handler is evaluated before the page and flow handlers, and the event is consumed when that first handler is called. The later custom event is a separate requirement: parameter-level reprompt handlers do not allow custom events, while a handler on the current page can handle a system-invoked custom event.
A - Incorrect: The expected postal-code reprompt is correct, but parameter-level event handlers are limited to invalid input during form filling and do not allow custom events. The custom shipping.escalate handler must use a supported page or flow scope.
B - Incorrect: A custom-event handler on the current page is supported. However, the page no-match fulfillment does not take precedence over an applicable reprompt handler for the parameter currently being filled. The parameter handler consumes this no-match event first.
C - Incorrect: An active-flow handler can handle a custom event, but the flow no-match fulfillment is not the selected invalid-input response here. Applicable parameter-level and then page-level handlers are evaluated before flow-level handlers, and the parameter handler consumes the event.
D - Correct: The current parameter's reprompt handler runs first and consumes the no-match event, producing the postal-code prompt. Page-Address-A is current and can host the separate custom-event handler, avoiding the unsupported parameter-level custom-event scope.

4.A Dialogflow CX architect reviews this failure-test fixture for Page-Checkout-A. The current-page SubmitOrder route invokes OrderBackend-A and has Page-Receipt-A as its configured transition target. Page-Checkout-A has an in-scope webhook.error.timeout handler whose fulfillment says, "Submission could not be confirmed. Try again." That event handler has no transition target. There are no other webhook error handlers or transitions in this fixture.

```
Initial current page: Page-Checkout-A
Matched intent: SubmitOrder
Selected route: SubmitOrder on Page-Checkout-A
Webhook outcome: timeout
Final current page: Page-Receipt-A
Timeout-handler fulfillment: not called
```

Which diagnosis and configuration change makes the configured timeout-handler response eligible on the failing webhook path?
A. The original route's target suppresses webhook error handling; remove the SubmitOrder route's transition target and retain the timeout handler on Page-Checkout-A.
B. The timeout event is too specific; replace its handler with webhook.error and retain Page-Receipt-A as the SubmitOrder route's target.
C. The timeout handler needs wider scope; move it to the active flow and retain Page-Receipt-A as the SubmitOrder route's target.
D. The timeout handler needs its own destination; give it a Page-Retry-A target and retain Page-Receipt-A as the SubmitOrder route's target.
Answer: A

Explanation:
The documented exception concerns the route that invokes the failing webhook. When that route already defines a transition target, the webhook can fail silently and the webhook error event is not invoked. Reaching the receipt page therefore does not confirm that the backend operation succeeded. Removing that original route target leaves the current-page timeout handler eligible to handle the timeout; changing the event type, scope, or the event handler's own target does not remove the suppressing condition.
A - Correct: The SubmitOrder route's configured target explains both the transition to Page-Receipt-A and the missing timeout response. Removing that target allows the timeout event to invoke the existing in-scope handler on the still-current checkout page. This makes the failure response eligible without treating page navigation as proof of backend success.
B - Incorrect: A generic webhook.error handler is not a workaround for a transition target on the original webhook-invoking route. The generic error event also requires that original route to have no transition target, and it is a fallback when no matching fine-grained error handler exists.
C - Incorrect: The original route's transition target still suppresses webhook error-event invocation. Moving the handler from the current page to the active flow does not change that condition, so broader scope does not address this failure path.
D - Incorrect: The target that matters is on the original route calling the webhook. Adding a transition target to the timeout handler does not make its event occur; that handler cannot apply its new target when its event was suppressed.

5.A Dialogflow CX architect is designing Page-Booking-A. A reservation request needs two values: room_id and booking_date. The business requires the reservation backend to be called only after both values have been collected, and collection of the final value should be sufficient to start the request without another reservation-intent utterance. During collection a user may say "Reserve it now" while booking_date is still missing. The architect must encode the invocation boundary in the workflow configuration. Assume the backend and extraction types are already configured correctly.
Which plan meets the requirement?
A. Collect room_id and booking_date through instructions instead of a required form; invoke the reservation webhook when the agent's response says collection is complete.
B. Make room_id and booking_date required form parameters; invoke the reservation webhook from a condition-only current-page route that requires the form's required parameters to be filled.
C. Make room_id and booking_date required form parameters; invoke the reservation webhook from a condition-only current-page route when room_id is filled, and continue collecting booking_date afterward.
D. Make room_id and booking_date required form parameters; invoke the reservation webhook from a ReserveRoom intent route with no condition, and instruct the agent to ask for missing values.
Answer: B
Explanation:
Required form parameters represent the collection state, and a condition-only route can use completion of that state as the trigger for fulfillment. This provides the configured boundary before a webhook invokes the reservation backend and does not require a fresh intent match. A prompt asking for missing information, a statement that collection is complete, or a condition on only one field does not supply the required two-field gate.
A - Incorrect: This plan makes the boundary depend on an instruction-led response rather than the required parameters' configured state. A generated claim that collection is complete is not the stated

two-field gate, so this plan does not encode the required transaction-invocation boundary.
B - Correct: The two required parameters establish what must be collected, and the route checks that their collection is complete before calling fulfillment. Because the route is condition-only and belongs to the current page, filling the final value can trigger it without another reservation intent. The gate controls invocation; it does not claim that the backend transaction has succeeded.
C - Incorrect: A filled room_id is insufficient when booking_date is also required. This route can invoke the backend before the date is present, then collect the remaining field after the invocation boundary has already been crossed.
D - Incorrect: A required form does not add an unstated condition to every intent route. The supplied ReserveRoom route has no completion condition, so the premature reservation utterance can match its intent while booking_date is missing. Instructions to ask for missing values do not repair that configured invocation path.