Practice With Microsoft AI-103 Mock Test

Shared by Watson on 17-06-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

You are deploying a support agent that enables users to upload photos.

You need to automatically classify uploaded images for harmful content. The solution must block content based on severity levels.

What should you do?

Options:

A- Implement image moderation.
B- Enable prompt shields.
C- Apply keyword scanning to optical character recognition (OCR) output by using Azure Vision in Foundry Tools.
D- Use blocklists.

Answer:

A

Explanation:

The correct answer is A. Implement image moderation. Azure AI Content Safety provides image analysis that classifies uploaded images for harmful content, including harm categories such as hate, sexual content, violence, and self-harm. Microsoft's Content Safety overview states that the Analyze Image API scans images for harmful content with multi-severity levels, which directly matches the requirement to automatically classify uploaded photos and block content based on configured severity thresholds.

Prompt Shields are intended to detect prompt injection and jailbreak-style attacks against generative models, not to classify image harm categories. Keyword scanning OCR output would only detect visible text extracted from the image and would miss visual harm in the image itself. Blocklists can help match known words or custom patterns, but they are not a complete image safety classifier and do not provide the built-in severity-based image harm classification required here. Image moderation is therefore the correct control for user-uploaded photos. Reference topics: Azure AI Content Safety, image moderation, harm categories, severity levels, Foundry guardrails, and responsible AI controls.

Question 2

Question Type: MultipleChoice

You have a Microsoft Foundry project named Project1 that contains an agent. The agent uses an OpenAPI 3.0 specification to call an external weather service.

The weather service requires a key to be passed in an HTTP header. The key value is stored as a connection in Project1.

You need to ensure that the key value from the connection is included automatically whenever the OpenAPI tool is invoked.

What should you configure in the OpenAPI specification?

Options:

A- an Azure Key Vault connection
B- a header parameter defined for each operation
C- an API key security scheme
D- a Bearer token security scheme

Answer:

C

Explanation:

The correct configuration is an API key security scheme. For Microsoft Foundry Agent Service OpenAPI tools, the OpenAPI specification must declare authentication through the components.securitySchemes section and use a scheme of type apiKey when the external service expects a key in a header. Microsoft's OpenAPI tool guidance states that API key authentication requires updating the OpenAPI spec security schemes with one scheme of type apiKey, and the tool then uses the associated project connection to supply the key value at runtime. This allows the key stored in Project1's connection to be injected automatically when the tool is invoked.

A header parameter defined separately for each operation is not the correct approach because credentials should not be modeled as ordinary operation parameters. The Foundry guidance explicitly indicates that parameters requiring the API key should be removed from the OpenAPI spec because the API key is stored and passed through a connection. A Bearer token security scheme is used for bearer-token-style authorization, not a generic weather API key passed in a custom HTTP header. Azure Key Vault is a secret store, but the scenario already stores the key in a Foundry project connection. Reference topics: Microsoft Foundry Agent Service, OpenAPI tools,

project connections, API key authentication, and OpenAPI security schemes.

Question 3

Question Type: MultipleChoice

You have a Microsoft Foundry project that ingests scanned PDF invoices stored in Azure Blob Storage. Each invoice contains printed line items and has a table-based layout.

Extracted results are stored as structured JSON and used as grounding data for an agent in a Retrieval Augmented Generation (RAG) solution.

You need to create a single analyzer that meets the following requirements:

* Extracts the invoice number, invoice date, vendor name, and total amount across varying templates * Returns confidence scores so that results with confidence below 0.80 can be routed for supervisor review

What should you use?

Options:

A- the Azure Content Understanding in Foundry Tools prebuilt-layout analyzer
B- a Foundry agent that has groundedness guardrails enabled to extract invoice fields and confidence scores
C- a custom Azure Content Understanding in Foundry Tools analyzer that defines the required fields as the extracted fields and the returned confidence scores for routing
D- the Azure Content Understanding in Foundry Tools prebuilt-documentSearch analyzer and search.score from the Azure AI Search results for routing

Answer:

C

Explanation:

The correct answer is C because the requirement is structured field extraction from invoices across varying templates, not only OCR or layout preservation. Azure Content Understanding analyzers are reusable configurations that combine content extraction, AI-powered analysis, and structured data output, and Microsoft states that custom analyzers can be created for specific extraction needs. In this case, the analyzer schema should define fields such as invoice number, invoice date, vendor name, and total amount so the output can be returned as structured JSON

for downstream RAG grounding.

The confidence-routing requirement also points to Content Understanding field confidence scores. Microsoft documentation states that every field can include a confidence score from 0 to 1, and that confidence scores can be used to automate high-confidence results while routing low-confidence results for human review. A threshold such as 0.80 is therefore an application routing rule based on the returned field confidence. The prebuilt-layout analyzer preserves layout but does not define invoice-specific business fields. Groundedness guardrails evaluate generated answers, not invoice field extraction. Azure AI Search search.score measures retrieval relevance, not extraction confidence. Reference topics: Content Understanding custom analyzers, document field extraction, structured JSON output, confidence scoring, and RAG grounding.

Question 4

Question Type: MultipleChoice

You are planning a Microsoft Foundry project named Project1 that will contain multiple agents. Each agent will access the same Azure AI Search resource.

You need to recommend a solution to centrally manage the Azure AI Search credentials within Project1. The solution must be implemented across all the agents.

What should you recommend?

Options:

A- Enable role-based access control (RBAC) for the Azure AI Search resource.
B- Add a connection to the Azure AI Search resource.
C- Disable key-based access control on the Azure AI Search resource.
D- Create a managed private endpoint that connects to the Azure AI Search resource.

Answer:

B

Explanation:

The correct recommendation is B. Add a connection to the Azure AI Search resource. Microsoft Foundry project connections are used to centrally define access from a project to external resources, including Azure AI Search. The official connection guidance states that you can add a connection by selecting an external service such as Azure AI Search, choosing the resource, and selecting the authentication method for that resource. This creates a reusable project-level

configuration rather than requiring each agent to store or duplicate search credentials independently.

For Foundry agents that use Azure AI Search, the Azure AI Search tool requires a project_connection_id, which is the resource ID of the project connection to Azure AI Search. This allows multiple agents to reference the same managed connection while using the configured endpoint and authentication settings consistently. RBAC may be part of a keyless authentication design, but it does not by itself create a centrally managed Foundry credential configuration. Disabling key-based access improves security posture but does not connect the agents. A managed private endpoint addresses network isolation, not credential centralization. Reference topics: Microsoft Foundry project connections, Azure AI Search tool, project connection IDs, authentication, and agent grounding.

Question 5

Question Type: MultipleChoice

You have a Microsoft Foundry project that contains an agent. The agent uses Azure Speech in Foundry Tools.

You fine-tune a baseline speech to text model for the en-us locale and publish the model.

The agent calls the Speech to text REST API and returns an error message indicating that the project ID is invalid.

You need to set the project property to the correct ID.

To what should you set the project property?

Options:

A- the custom speech endpoint URL
B- the project URL
C- the project ID
D- the custom speech project ID

Answer:

D

Explanation:

The correct answer is D. the custom speech project ID. For custom speech fine-tuning, the Speech to text REST API uses a project property that must refer to the Custom Speech project, not the general Microsoft Foundry project. Microsoft's Custom Speech guidance states that when using the Speech to text REST API for custom speech, you must set the project property to the ID of your custom speech project. It also explicitly notes that the custom speech project ID is not the same as the Microsoft Foundry project ID.

This distinction explains the invalid project ID error. Supplying the Foundry project ID, project URL, or endpoint URL does not identify the Custom Speech project that owns the fine-tuned speech model. The custom speech endpoint URL is used when calling a deployed custom model endpoint for recognition, but it is not the value of the REST API project property. The project URL is also not accepted because the API expects the identifier value. Reference topics: Azure Speech in Foundry Tools, Custom Speech fine-tuning, Speech to text REST API, custom speech project ID, model publication, and endpoint configuration.

Question 6

Question Type: MultipleChoice

You have a Microsoft Foundry project that generates product marketing images from text prompts.

After publishing several images, the legal team at your company identifies a competitor's logo on a sign in the background of an image.

You need to remove only the logo, while preserving the rest of the image.

What should you do?

Options:

A- Increase the prompt guidance strength.
B- Modify the original prompt to exclude brand names.
C- Apply a mask-based inpainting edit to the part of the image that contains the logo.
D- Rerun the prompt by using a different random seed.

Answer:

C

Explanation:

The correct answer is C because the requirement is a localized image edit: remove only the competitor logo while preserving the rest of the already generated image. Azure OpenAI image editing is designed for modifying existing images based on a text instruction, rather than regenerating the entire image from scratch. Microsoft's Azure OpenAI image guidance states that the Image Edit API modifies existing images and requires an input image as part of the request. In a mask-based inpainting workflow, the mask identifies the exact region to change, allowing the model to replace only the logo area while retaining surrounding background, composition, lighting, and product content.

Increasing prompt guidance strength would affect adherence during generation, but it would not safely remove a specific logo from a completed image. Modifying the original prompt and regenerating may create a different image and does not guarantee preservation of the approved visual content. Rerunning with a different random seed also changes the image unpredictably and may introduce new brand or legal issues. Mask-based inpainting is the minimal-change remediation method for post-generation brand cleanup. Reference topics: Azure OpenAI image editing, inpainting, mask-guided edits, image generation governance, and computer vision solutions.