Prepare Smart for Success Free Oracle 1Z0-1157-26 Exam Questions and Answers

Ready to pass faster? Grab free and updated Oracle Agentic AI Foundations Associate exam PDF questions now. Get authentic 1Z0-1157-26 dumps packed with verified answers and secure your certification success with PrepBolt 1Z0-1157-26 exam pdf questions and answers.

Thank you for Downloading 1Z0-1157-26 exam PDF Demo

https://prepbolt.com/1Z0-1157-26.html

QUESTIONS & ANSWERS

DEMO VERSION

(LIMITED CONTENT)

Question 1

Question Type: MultipleChoice

How are tool calls handled between the LLM and the application?

Options:

A- It performs the external operation directly during inference.
B- It allows tools to run without application control.
C- It generates a tool-call request for the application to validate and run.
D- It grants the orchestration layer unrestricted system access.

Answer:

C

Explanation:

For application-defined function tools, an LLM does not inherently execute the external operation itself. Instead, the model produces a structured tool-call request identifying the selected function and supplying arguments. The application or agent runtime then interprets that request, applies appropriate validation or authorization, invokes the corresponding implementation, and returns the result to the model for subsequent reasoning.

The OpenAI Responses API defines function calls as custom tools supplied by the developer that enable the model to request execution of application code using typed arguments. The OpenAI Agents SDK makes the separation explicit: a FunctionTool contains the tool name, description, parameter JSON schema, and an on_invoke_tool implementation that actually executes when the runtime processes the model-generated arguments.

This boundary is critical for security. The model proposes an action; controlled application/runtime logic performs the action. The model is therefore not automatically granted direct database, filesystem, network, or operating-system privileges merely because a tool has been described to it.

Options A, B, and D incorrectly remove this enforcement boundary. Consequently, C is the correct architectural description and matches the supplied question source.

Study Guide reference/topic: OpenAI Responses API and Agents SDK --- function calling, structured tool calls, application execution, validation, schemas, and security boundaries.

Question 2

Question Type: MultipleChoice

When is MCP most valuable?

Options:

A- For simple local utilities like add() and multiply().
B- When building a single-agent app with no external dependencies.
C- When defining prompts and output parsers inside a single LangChain chain.
D- When tools are external or cloud-hosted.

Answer:

D

Explanation:

MCP delivers its greatest architectural value when an AI application must interact with capabilities that exist outside the application's own process, particularly external services, databases, SaaS platforms, APIs, or cloud-hosted tools. Instead of implementing a proprietary integration contract for every agent/tool combination, an MCP server exposes those capabilities using a standardized protocol.

The MCP specification describes the protocol as a standardized mechanism for integrating LLM applications with external data sources and tools. Its tools specification further states that MCP servers can expose capabilities that query databases, call APIs, perform computations, or otherwise interact with external systems.

For trivial local functions such as add() or multiply(), a normal in-process function-tool definition is usually simpler because no separate server protocol is needed. Similarly, an application that has no external dependencies gains relatively little from introducing MCP solely for prompts or local output parsers.

Thus the decisive use case is integration across system or deployment boundaries, especially where capabilities should be reusable by multiple MCP-compatible clients.

Therefore, D is the correct answer and is also explicitly marked correct in the uploaded source.

Study Guide reference/topic: Model Context Protocol (MCP) Fundamentals --- external tools, MCP servers, interoperability, reusable integrations, and remote services.

==========

Question 3

Question Type: MultipleChoice

Why are docstrings especially important when defining LangChain tools with the @tool decorator?

Options:

A- They determine which API provider LangChain uses.
B- They improve Python execution speed.
C- They encrypt tool arguments before execution.
D- They tell the LLM when and how to use the tool.

Answer:

D

Explanation:

When LangChain's @tool decorator is applied to a Python function, the function's documentation becomes part of the metadata shown to the language model. LangChain explicitly states that, by default, the function's docstring becomes the tool description, helping the model understand when the capability should be selected.

This metadata is operationally significant because tool selection is model-driven. A clear description communicates the tool's purpose, appropriate usage conditions, expected arguments, and semantic boundaries. LangChain's context-engineering guidance emphasizes that tool names, descriptions, argument names, and argument descriptions guide the model's reasoning about when and how a particular tool should be invoked.

Docstrings therefore affect agent reliability, not Python runtime performance. They do not determine which underlying model provider is used, and they have no cryptographic role in protecting function parameters. Poor or ambiguous descriptions can cause an LLM to choose an inappropriate tool or supply unsuitable arguments even when the Python implementation itself is technically correct.

Consequently, D precisely captures why docstrings are particularly important for @tool-defined LangChain functions. The uploaded source confirms the same answer.

Study Guide reference/topic: LangChain for AI Agents --- @tool decorator, tool descriptions, docstrings, schemas, tool selection, and context engineering.

==========

Question 4

Question Type: MultipleChoice

Which OCI capability is required for serving fine-tuned or imported custom models?

Options:

A- Free Tier compute instances
B- Dedicated AI Clusters
C- Shared On-Demand inference
D- OCI Object Storage buckets

Answer:

B

Explanation:

OCI Generative AI uses Dedicated AI Clusters to provide the isolated compute infrastructure required for fine-tuning and hosting custom model workloads. Oracle defines Dedicated AI Clusters as compute resources dedicated to a customer's models rather than shared with other tenancies. They can be created specifically for fine-tuning or for hosting model endpoints.

Oracle's current model onboarding workflow confirms the requirement. For imported models, the process includes importing the model, creating a hosting Dedicated AI Cluster, creating an endpoint, and then invoking the model. Fine-tuned models similarly require dedicated clusters for fine-tuning and subsequent hosting.

Shared On-Demand inference is appropriate for supported Oracle-hosted pretrained models, but it does not provide the dedicated isolated serving environment required by these custom model workflows. Object Storage can be an input location for model artifacts or training data, but it is storage rather than model-serving infrastructure. General-purpose Free Tier compute is likewise not the managed Generative AI capability Oracle specifies for custom-model serving.

Thus, B is correct and agrees with the uploaded course material.

Study Guide reference/topic: OCI Enterprise AI Agents --- Dedicated AI Clusters, imported models, fine-tuned custom models, hosting clusters, and endpoints.

==========

Question 5

Question Type: MultipleChoice

Which value associates OCI Responses API requests with a specific OCI Generative AI Project?

Options:

A- Object Storage bucket name
B- Tenancy display name
C- VCN OCID
D- Project OCID

Answer:

D

Explanation:

OCI Responses API requests are associated with an OCI Generative AI Project through the project's OCID --- Oracle Cloud Identifier. Oracle requires an OCI Generative AI project for agent-related OpenAI-compatible API calls and uses the project identifier to determine the project context under which responses, conversations, files, containers, retention settings, and related resources operate.

Oracle's OCI Responses API documentation shows the OpenAI client configured with a project parameter containing a Generative AI Project OCID. Oracle explicitly states that this value identifies the OCI Generative AI project for the request. Oracle's project documentation further explains that projects organize agent-specific artifacts, provide isolation boundaries, and that the project OCID must be referenced in API and SDK calls to apply project settings during runtime.

An Object Storage bucket could contain data used by another workflow but does not identify the Generative AI project. The tenancy display name identifies a tenancy conceptually but not the target project. A VCN OCID refers to network infrastructure.

Therefore, D is correct and matches the uploaded answer key.

Study Guide reference/topic: OCI Enterprise AI Agents --- Generative AI Projects, Project OCID, OCI Responses API configuration, and project isolation.

==========

Question 6

Question Type: MultipleChoice

What is the purpose of the Vector Stores API?

Options:

A- Translating text between languages.
B- Encrypting Object Storage buckets.
C- Streaming video into AI agents.
D- Indexing and retrieving data by meaning.

Answer:

D

Explanation:

The Vector Stores API provides infrastructure for ingesting content and retrieving the portions that are semantically relevant to a query. Files associated with a vector store can be chunked and prepared for vector-based retrieval, allowing applications and agents to locate content based on semantic similarity rather than only exact lexical matches.

OpenAI's official Vector Stores API supports searching a vector store with a natural-language query and returns relevant content chunks together with similarity scores. The API also supports attaching files to vector stores and configuring the chunking strategy used during ingestion. This architecture is foundational to retrieval-augmented generation and File Search workflows: source material is indexed, a user query retrieves semantically related chunks, and those chunks can then provide grounded context to a model.

Language translation is a generative-model task. Object Storage encryption is a cloud-storage security function, while video streaming is unrelated to the purpose of a vector store.

Accordingly, ''Indexing and retrieving data by meaning'' is the technically correct description and is also the answer specified by the uploaded question source.

Study Guide reference/topic: OpenAI Responses API and Agents SDK --- Vector Stores, semantic retrieval, chunking, File Search, similarity ranking, and RAG.

==========

Thank You for trying 1Z0-1157-26 PDF Demo

To try our 1Z0-1157-26 practice exam software visit link below

https://prepbolt.com/1Z0-1157-26.html

Start Your 1Z0-1157-26 Preparation

Use Coupon “SAVE50” for extra 50% discount on the purchase of Practice Test Software. Test your 1Z0-1157-26 preparation with actual exam questions.