CERTLABZ CERTIFICATION BOARD

Certified Prompt Engineering Architect (CPEA)

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 prompt engineering architect 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: Prompt Engineering Foundations

CONFIG

Sarah Miller drafts a reusable prompt template with a placeholder for the input.

```
Role: You are a helpful tutor.
Task: Explain {topic} simply.
Style: Two short sentences.
Audience: A new beginner.
```

Listing 1.1 A prompt template.

What does the {topic} part represent?

A. A model weight
B. A fixed output
C. A variable slot
D. A stop token

Correct answer: C

The braces mark a variable slot that is filled with real input each time the template runs. It is not a fixed output, a model weight, or a stop token, all of which serve other purposes.

Question 2 Domain 2: Prompting Techniques DATA

James Carter compares prompting techniques, with one description missing.

Technique	What it does
Zero shot	No examples given
Few shot	(to choose)
Chain of thought	Shows step reasoning
Role prompt	Sets a persona

Table 2.1 Techniques and meaning.

What does few shot prompting do?

A Adds sample examples
B Hides all context
C Trains new weights
D Blocks the output

Correct answer: A
Few shot prompting adds a small set of sample examples so the model learns the pattern from them. It does not retrain weights, hide context, or block output, which are unrelated to giving examples.

Question 3 Domain 3: Working with Language Models CONFIG

Emma Wilson reviews the message list sent to a chat model.

```
[
  {role: system, content: rules},
  {role: user, content: question},
  {role: assistant, content: reply}
]
```

Listing 3.1 A message list.

Which role sets the overall behaviour?

A user
B system
C assistant
D function

Correct answer: B
The system role carries the rules and sets the overall behaviour for the whole exchange. The user role holds the question, the assistant role holds the reply, and function is not shown here.

Question 4 Domain 4: Retrieval and Tools CONFIG

Olivia Davis defines a JSON schema for a tool the model may call.

```
{
  name: get_weather,
  params: { city: string },
  returns: { tempC: number }
}
```

Listing 4.1 A tool schema.

Why give the model this schema?

A To store its memory

B To pick the model

C To call the tool right

D To hide the prompt

Correct answer: C
The schema tells the model the tool name and inputs so it can call the tool right with valid arguments. It does not store memory, pick a model, or hide the prompt.

Question 5 Domain 5: Evaluation and Testing DATA

Liam Anderson scores two prompt versions on a fixed test set.

Prompt	Pass rate	Cost
Version A	72%	Low
Version B	88%	Low
Version C	74%	High
Version D	60%	Low

Table 5.1 Prompt scores on the test set.

Which prompt should the team ship?

A Version C

B Version D

C Version B

D Version A

Correct answer: C
Version B has the highest pass rate at a low cost, so it is the strongest choice to ship. The others score lower, and Version C matches none of that quality while costing more.

Question 6 Domain 6: Safety, Governance and Cost CONFIG

Ava Thompson lists guardrail settings applied before a prompt reaches users.

```
input_filter:  block unsafe text
max_tokens:    cap the output
pii_redact:    mask personal data
log_review:    proctored by staff
```

Listing 6.1 Guardrail settings.

Which setting limits the response length?

A pii_redact
B input_filter
C max_tokens
D log_review

Correct answer: C
The max_tokens setting caps the output, which limits the response length and helps control cost. The input filter blocks unsafe text, pii_redact masks personal data, and log review is a proctored human check.

Question 7 Domain 1: Prompt Engineering Foundations

David Brown writes a prompt that says be helpful but never states the format, the length, or who the reader is, and the replies vary wildly. What is the best fix?

A Add clear instructions
B Raise the temperature
C Send a longer prompt
D Retrain the model

Correct answer: A
Clear instructions on format, length, and audience give the model the context it needs to produce steady replies. Raising temperature adds randomness, mere length does not help, and beginners do not retrain the model.

Question 8 Domain 3: Working with Language Models

A team notices that setting a high temperature makes answers to a factual lookup change every run, which they do not want. What best steadies the output?

A Lower the temperature
B Add more examples
C Remove the system role

D Grow the token cap

Correct answer: A
A lower temperature makes the model pick the most likely tokens, so factual answers stay steady across runs. Adding examples, growing the token cap, or dropping the system role does not control that run to run variation.

Question 9 Domain 4: Retrieval and Tools

A chatbot invents facts about a companys refund rules that are not in any prompt. The team wants answers grounded in real policy text. What approach fits best?

A Retrieve the policy
B Raise the temperature
C Shorten the prompt
D Add more personas

Correct answer: A
Retrieving the real policy text and placing it in the prompt grounds the answer in source material and curbs invented facts. Higher temperature, shorter prompts, or extra personas do not supply the missing policy.

Question 10 Domain 5: Evaluation and Testing

A team ships a new prompt after one lucky manual try and later finds it fails on many real inputs. What practice would have caught this earlier?

A A shorter prompt
B A fixed test set
C A higher token cap
D A second persona

Correct answer: B
Running the prompt against a fixed test set of varied inputs measures quality before release and catches failures a single try misses. A shorter prompt, larger token cap, or extra persona does not test broad coverage.

Published by the CertLabz Certification Board, a division of CertLabz LLC. Companion to objectives document CPEA-001. https://certlabz.com/certifications/certified-prompt-engineering-architect.html