

# Updated CT-GenAI Testkings & Reliable Positive CT-GenAI Feedback Promise you "Money Back Guaranteed"

## CT-GenAI ISTQB® Exam

P.S. Free & New CT-GenAI dumps are available on Google Drive shared by Lead1Pass: <https://drive.google.com/open?id=1Ur9A5IE-BpMiQ0GJDseLkRk1vSINiWzd>

If you have problems with your installation or use on our CT-GenAI training guide, our 24 - hour online customer service will resolve your trouble in a timely manner. We dare say that our CT-GenAI preparation quiz have enough sincerity to our customers. You can free download the demos of our CT-GenAI Exam Questions which present the quality and the validity of the study materials and check which version to buy as well.

Our CT-GenAI study braindumps are comprehensive that include all knowledge you need to learn necessary knowledge, as well as cope with the test ahead of you. With convenient access to our website, you can have an experimental look of free demos before get your favorite CT-GenAI prep guide downloaded. You can both learn useful knowledge and pass the exam with efficiency with our CT-GenAI Real Questions easily. We are on the way of meeting our mission and purposes of helping exam candidates to consider the exam as a campaign of success and pass the exam successfully.

>> Updated CT-GenAI Testkings <<

### CT-GenAI Real Study Dumps Would be a Reliable Exam Questions for You

Everyone wants to stand out in such a competitive environment, but they don't know how to act. Maybe our CT-GenAI exam questions can help you. Having a certificate may be something you have always dreamed of, because it can prove that you have a certain capacity. Our CT-GenAI learning materials can provide you with meticulous help and help you get your certificate. Our CT-GenAI training prep is credible and their quality can stand the test. Therefore, our CT-GenAI practice materials can help you get a great financial return in the future and you will have a good quality of life.

### ISQI ISTQB Certified Tester Testing with Generative AI (CT-GenAI) v1.0 Sample Questions (Q34-Q39):

#### NEW QUESTION # 34

Which setting can reduce variability by narrowing the sampling distribution during inference?

- A. Lowering temperature
- B. Using a larger context window
- C. Increasing learning rate
- D. Increasing temperature

**Answer: A**

Explanation:

In the context of LLM inference, Temperature is a hyperparameter that controls the randomness or "creativity" of the model's output. When the temperature is set high, the model's probability distribution is "flattened," meaning it is more likely to select less-probable tokens, leading to more diverse and sometimes unpredictable text. For software testing, where precision and repeatability are paramount, lowering the temperature (Option C) is the standard practice. A temperature of 0.0 makes the model "deterministic," meaning it will consistently choose the token with the highest probability. This narrows the sampling distribution and significantly reduces variability between runs. While a larger context window (Option D) allows the model to process more information, it does not directly control the randomness of token selection. Similarly, the "learning rate" (Option B) is a parameter used during the training or fine-tuning phase, not during inference. For generating test cases or scripts that must follow strict logic, a lower temperature ensures that the model remains focused and produces consistent results.

### NEW QUESTION # 35

Who typically defines the system prompt in a testing workflow?

- A. A tester configuring the assistant
- B. CI server automatically without human input
- C. Product owner in user stories only
- D. End user during normal chat use

**Answer: A**

Explanation:

In professional Generative AI applications, the system prompt (sometimes called the system message) is the foundational set of instructions that defines the AI's persona, boundaries, and overall behavior. In a testing workflow, this is typically defined by a tester or test engineer who is configuring the AI assistant for a specific project. Unlike the user prompt, which changes with every interaction, the system prompt remains relatively static and acts as a "guardrail" to ensure the model stays in its role (e.g., "You are an expert in ISO 26262 automotive testing standards"). By defining the system prompt, the tester ensures that the model consistently uses specific terminology, adheres to data privacy constraints, and formats its output according to the team's requirements. While end users (Option B) provide the task-specific input, they do not usually have the permissions or technical need to alter the underlying system-level instructions. Similarly, while CI servers (Option C) might trigger the prompt, they do not "define" the human-centric logic contained within it. Properly crafting the system prompt is a core part of setting up an AI-augmented test environment.

### NEW QUESTION # 36

What is a hallucination in LLM outputs?

- A. A systematic preference learned from data
- B. A logical mistake in multi-step deduction
- C. A transient network failure during inference
- D. Generation of factually incorrect content for the task

**Answer: D**

Explanation:

A hallucination refers to a phenomenon where a Large Language Model generates text that is grammatically correct and seemingly plausible but is factually incorrect or unsupported by the provided context or real-world data. In the context of software testing, this is a critical limitation. For example, an LLM might generate a test case for a software feature that does not exist or cite a non-existent API parameter. These errors occur because LLMs are probabilistic engines designed to predict the "most likely" next token rather than "reasoning" from a set of verified facts. They do not have a built-in "truth" mechanism. While a logical mistake (Option B) is a failure in reasoning and a systematic preference (Option D) describes bias, a hallucination is specifically about the fabrication of

information. Testers must be particularly vigilant regarding hallucinations, as they can lead to "false confidence" in test coverage or the creation of invalid bug reports. Mitigations include grounding the model with Retrieval-Augmented Generation (RAG) and implementing rigorous "human-in-the-loop" verification of all AI-generated test artifacts.

### NEW QUESTION # 37

A prompt begins: "You are a senior test manager responsible for risk-based test planning on a payments platform." Which component is this?

- A. Context
- B. Constraints
- C. Instruction
- **D. Role**

**Answer: D**

Explanation:

In structured prompt engineering, theRolecomponent (also known as a Persona) is used to set the perspective, expertise, and tone of the LLM's response. By assigning the role of a "senior test manager," the tester instructs the model to adopt the specific domain knowledge, vocabulary, and professional standards associated with that position. This technique is highly effective because LLMs are trained on vast datasets containing diverse professional documents; invoking a specific persona helps the model narrow its "latent space" to retrieve information relevant to that specific field. For instance, a senior test manager persona will prioritize risk management, resource allocation, and high-level strategy, whereas a "junior developer" persona might focus more on syntax and local unit tests. WhileContext(Option B) provides the background of the project andInstruction(Option A) defines the specific task to be performed, theRoleserves as the foundation for how those instructions are interpreted. This ensures the generated testware aligns with the expected professional seniority and organizational maturity required for high-stakes environments like a payments platform.

### NEW QUESTION # 38

In the context of software testing, which statements (i-v) about foundation, instruction-tuned, and reasoning LLMs are CORRECT?

- i. Foundation LLMs are best suited for broad exploratory ideation when test requirements are underspecified.
- ii. Instruction-tuned LLMs are strongest at adhering to fixed test case formats (e.g., Gherkin) from clear prompts.
- iii. Reasoning LLMs are strongest at multi-step root-cause analysis across logs, defects, and requirements.
- iv. Foundation LLMs are optimal for strict policy compliance and template conformance.
- v. Instruction-tuned LLMs can follow stepwise reasoning without any additional training or prompting.

- A. i, ii, iii (Duplicate entry in original source)
- **B. i, ii, iii**
- C. ii, iii, iv
- D. i, iii, v

**Answer: B**

Explanation:

Understanding the hierarchy of LLM types is vital for selecting the right tool for specific testing tasks.

Foundation LLMs are trained on massive datasets to predict the next token; they excel at broad, creative

"ideation" (Statement i) but often struggle with following specific instructions or constraints (making Statement iv

incorrect). Instruction-tuned LLMs have undergone additional training (Fine-tuning) to follow explicit commands and templates. They are highly effective at structured tasks like converting requirements into Gherkin feature files (Statement ii). Reasoning LLMs (or those

utilizing specialized prompting like Chain-of-Thought) are designed to handle complex, multi-stage logic. This makes them the

superior choice for diagnostic tasks like root-cause analysis, where the model must synthesize information across logs and

requirements to find a defect's origin (Statement iii). Statement v is incorrect because while instruction-tuned models are capable,

complex "stepwise reasoning" usually requires specific prompting techniques or the inherent logic of specialized reasoning models.

Therefore, the combination of i, ii, and iii represents the correct alignment of model capability to testing functionality.

### NEW QUESTION # 39

.....

