

The generative AI lifecycle

A generative AI system has two major phases.

Process • “Training data → Model training → Foundation model → Prompt → Generated response”

Two phases

- Training: The model learns patterns from large datasets.
- Inference: The trained model generates an output for a new input.

Key distinction

- Training: Creates or updates the model
- Inference: Uses the model

Exam focus

If a deployed model answers a user, classifies content, or generates text, the model is performing inference.

Foundation models

Answering questions

Summarizing content

Generating text or images

Foundation model

Extracting information

Following instructions

Generating code

Key message

A foundation model learns broad patterns from large, varied datasets. Developers adapt it through prompting, grounding, tools, or additional training.

Training, inference, and prediction

Term	What happens	Example
Training	The system adjusts model parameters by learning patterns from data	Learning relationships among words and concepts
Inference	The trained model processes a new input	Responding to a customer question
Prediction	The model estimates the most likely output	Selecting the next token

Callout

Repeat

Generative models repeatedly predict what content should come next.

Not retrieval

The model does not retrieve a complete prewritten answer from training data.

Exam trap

Prediction can mean estimating the next token—not forecasting a future event.

Tokens are the model's working units

Models process tokens, not raw sentences.

Example · “Generative AI is useful.”

Possible tokenization

Gener | ative | AI | is | useful | .

A token may represent

- A whole word
- Part of a word
- Punctuation
- A symbol
- A fragment of code

Callout

Token boundaries depend on the model's tokenizer.

Context windows define available working context

The context window limits how much information a model can consider during one inference request.

It may include

- System and developer instructions
- The user's prompt
- Conversation history
- Grounding data or retrieved documents
- Tool results
- The generated response

Key message

More context does not guarantee a better answer. Relevant context matters more than sheer volume.

Prompts become responses

A prompt provides the input and instructions for the model.

Prompt · “Summarize this support ticket in two sentences and identify the affected product.”

A prompt can specify

- The task
- Relevant context
- Constraints
- The requested output format
- Examples of acceptable results

Response

- Generated output: A short summary plus the extracted product name.

What the model does

The model produces a response by generating a sequence of output tokens.

Generated answers are probabilistic

For each output position, the model assigns probabilities to possible next tokens.

Example · “Your replacement order will arrive...”

Possible next token

Token	Probability
tomorrow	48%
Monday	27%
shortly	15%
after	10%

Consequences

- The same prompt can produce different answers.
- Fluent wording does not prove factual accuracy.
- The model can generate unsupported information.
- Model settings can change the amount of variation.

Exam trap

A polished response does not mean that the model verified every statement against a trusted source.