Prompt Engineering sits at the top of two major principles:

1. **PROVIDE CLEAR AND SPECIFIC INSTRUCTIONS TO THE MODEL**
2. **GIVE TIME TO THE MODEL TO THINK AND REASON**

PROVIDE CLEAR AND SPECIFIC INSTRUCTIONS TO THE MODEL

The effectiveness of an LLM is directly proportional to the clarity of your intent. Think of the model not as a mind-reader, but as a highly capable intern who lacks any prior context. If you give a vague dump of words, the model has to guess your goals and it will often guess wrong. By being specific, you guide the model's focus, reduce hallucinations, and ensure the output matches your desired tone, format, and length.
Remember: Specific doesn't always mean short. A longer prompt that provides clarity is significantly more effective than a cryptic one-liner.

There are several tactics which, when enforced, help to follow this principle; they are as follows:

Tactic 1: Use delimiters to clearly indicate distinct parts of the input

Delimiters can be anything like: ```, """, < >:
What these delimiters help in doing is that, we can specify exactly what we want to do with what part of the prompt. Eliminates hallucinations and clearly indicates to the model what it has to do.

For example: Translate the part of the prompt between triple backticks(```) into French and translate the part of the prompt between tags(<>) into Hindi.

Tactic 2: Ask for a structured output

Request the output from the model in a specific format. This will ensure that the output you receive is in the desired format and also the fact that all the information that you need is received.

For example: Return the output in JSON format.

Tactic 3: Ask the model to check if the conditions are satisfied

This will ensure that the output is sent only if the set of conditions are met. In case of missing conditions, the model will respond accordingly ensuring that it doesn’t respond wildly. It ensures that the model doesn’t send made up responses or hallucinates.

For example: "If the text provided contains a date and a location, summarize the event. If it does not contain both, write 'Information incomplete.'"
Input: "There is a meeting tomorrow."
Output: "Information incomplete."

Tactic 4: Few-Shot prompting

Provide the model with certain input-output examples. In this case, the model will understand the kind of response you are expecting and will respond accordingly; answers will be accurate and high-quality.

For example: Answer in a consistent style:
<Teacher>: etrha
<Student>: heart
<Teacher>: gunls
<Student>: lungs
<Teacher>: dinkey

Output: kidney

GIVE TIME TO THE MODEL TO THINK AND REASON

If a model is struggling to solve a complex problem, it might be because it is trying to generate the final answer too quickly. Just as a human might make a "silly mistake" if asked to solve a multi-step problem in their head instantly, LLMs can "hallucinate" or provide flawed logic when they aren't forced to work through the steps. By allowing the model to reason through its logic first, you ensure that the final output is based on a chain of thought rather than a lucky guess.

There are several tactics which, when enforced, help to follow this principle; they are as follows:

Tactic 1: Specify the steps required to complete a task
Instead of giving the model one big goal, you break the process down into a sequential checklist. This ensures the model doesn't skip critical logic.

For example:
"Follow these steps to respond: Summarize the user's complaint in one sentence. Check if the complaint mentions a refund. Based on the summary and refund status, draft a polite apology."

Tactic 2: Instruct the model to work out its own solution before rushing to a conclusion
Ask the model to double-check. Don't ask the model to tell if the user is right or wrong directly. First, let it solve the problem by itself and only then, compare to give the final result. In such situations, the chances of it making mistakes or hallucinating reduces.

For example: "A student says 2x+5=10, so x=10. Instructions: First, solve the equation step-by-step yourself. Only after you have your own answer, determine if the student is correct or incorrect."