Session 1 Handout: A Prompt Is an Assignment

FAST-03 Prompt Engineering for the Workplace · 7formations · Instructor: Wadood Chaudhary · All examples fictional

Why this session matters

Most disappointing AI output traces back to the prompt, not the tool. A generative model writes something new every time, and whatever you leave out, it guesses. At work, guesses become wrong dates, promises your business never made, and policies that don't exist. This session gives you a five-part card that turns a vague request into an assignment the model can do and you can check.

Key concepts

1. Search box vs. generative model

A search box **finds** pages that already exist. You judge the source. A generative model **writes** new text, predicting likely next words from patterns in its training and whatever you give it. That has two consequences:

- With no source, it fills gaps with plausible content. Plausible is not the same as true.
- The same prompt can produce different outputs on different days. Your check matters every time.

Some tools also search the web, but the answer is still written by the model and can still misstate what the pages say.

2. The five-part card

Part	Question it answers	Example (Cape Harbor Inn)
Role	Who is writing, and who is reading?	"I'm the guest-services lead writing to past guests."
Context	What is the only source?	The fact sheet, fenced between markers
Task	What is the one job?	"Write the cleanup announcement."
Format	What shape: length, structure, tone?	"120 words, three short paragraphs, friendly; end with how to sign up."
Check	What must be flagged, verified, or left out?	"If not in the sheet, write 'not in source'; list Gaps; no rain date or prices."

Naming note: the course design calls the last two parts "output" and "constraints." Format is the output; Check is the constraints. It's the same card.

One job per prompt. "Summarize this and draft a reply and make a flyer" is three jobs. Each gets done less well, and you can't check them separately. Split them.

3. Source binding

Source binding ties the model to text you trust:

1. **Fence it.** Put the source between clear markers (`---` or `<source>` tags). Instructions stay outside.

2. **Say "use only."** "Use only the fact sheet between the markers."
3. **Require "not in source."** Tell the model what to write when something is missing.
4. **Ask for a Gaps line.** "List anything a reader would need that the sheet doesn't answer."

The Gaps line is a gift: it becomes your question list for the person who owns the facts.

4. Why flattery doesn't help

"You are the world's best copywriter" gives the model nothing to check against. A specific role ("front-desk associate replying to a guest who stayed last summer") sets vocabulary, tone, and what the reader needs.

Worked example: rebuilding the thin prompt

Thin: "Write something nice about the cleanup so people come." *Output problems:* invented rain date (May 23), "free shuttle from the inn," "raffle prizes." None are on the sheet.

Rebuilt:

```
Role: I'm the guest-services lead at Cape Harbor Inn writing to past
guests who live on or visit the Cape.
Context: Use ONLY the fact sheet between the markers.
---
[fact sheet]
---
Task: Write an announcement inviting volunteers to the beach cleanup.
Format: 120 words, three short paragraphs, friendly and plain; end with
how to sign up and the May 13 deadline.
Check: Do not add facts that aren't on the sheet (no rain date, prizes,
or transport). If a reader would need something the sheet doesn't say,
write "not in source" and list it under "Gaps:".
```

Result: every fact traces to the sheet. Gaps: rain plan; parking. The events manager answers both before posting.

Checklist: before you press Enter

- ☐ Role names writer and reader.
- ☐ Source fenced; "use only" stated.
- ☐ One job.
- ☐ Format: length, structure, tone, ending.
- ☐ Check: "not in source," Gaps, named forbidden additions.
- ☐ Nothing in the prompt fails the shared-printer test.

Do and don't

Do	Don't
Give the model the source	Ask it to "look up" your own business facts

Do	Don't
Name the reader	Write "for everyone"
Name forbidden additions	Rely on "be accurate"
Split multi-job requests	Stack three jobs into one prompt
Route Gaps to the fact owner	Let a second prompt guess the gaps

Try this at work

1. Find a prompt you used last week (or a request you'd like to make).
2. Score it against the card: which parts are missing?
3. Rebuild it with a safe source (public page, approved template, or a fictional stand-in).
4. Run both versions in new chats and count facts not in the source.

Glossary

Term	Meaning
Generative model	AI that writes new text (or images) from patterns, rather than retrieving pages
Prompt	The instructions and material you give the model
Source binding	Restricting the model to text you supply
Fence	Markers around source text that separate it from instructions
Gaps line	A list of what the reader needs that the source doesn't answer
Thin prompt	A prompt missing most of the five parts

Key takeaways

- A prompt is an assignment: whatever you leave out, it guesses.
- Five parts: Role, Context, Task, Format, Check.
- Fence the source, say "use only," and ask for Gaps.