

TEACHER-FACING FRIDAY DESIGN

From Asking to Structuring: Designing AI Interaction

CAS Human-AI Collaboration | Module 1 | Unit 1

LITERACY	COLLABORATION MOVE	CONTACT TIME	CHALLENGE
Interaction	Instruction	365 minutes	Research-Backed Professional Artefact

Position in Module

- **Unit:** 1
- **Topic:** From Asking to Structuring: Designing AI Interaction
- **Literacy perspective:** Interaction
- **Collaboration level:** Instruction
- **Fundamentals focus:** Generative AI Fundamentals
- **Why this Friday matters:** This Friday develops the unit's collaboration move through the specified literacy lens and prepares participants for the associated evidence-based challenge.

Literacy Perspective

- **Primary literacy:** Interaction
- **Secondary literacy, if any:** None
- **What this literacy explains:** It explains the main dimension through which participants can predict, influence, and interpret AI behaviour in this unit.
- **What participants should manipulate, observe, and interpret:** Participants manipulate the unit's experiment focus, observe the stated outcomes, and interpret results through the literacy perspective.

Collaboration Level

- **Level relation:** Instruction
- **Collaboration move being learned:** Participants practice the unit's collaboration move as a transferable way of working with AI.
- **What participants should be able to do differently after this Friday:** They should be able to design a more controlled AI-supported interaction, process, decision arrangement, or workflow in their own context.

Generative AI Fundamentals

This Friday should begin with a basic introduction to what a large language model is and why interaction can control its behaviour. The goal is not technical completeness, but a usable mental model for the rest of the module.

- Start from a simple model such as the linear function $y = ax + b$. Use it to explain that a model has parameters, here a and b ; that training means specifying parameters from training data; that inference means inputting x and obtaining y ; and that y can be interpreted as a prediction, for example a classification if it lies above or below a threshold.
- **Transfer this idea to large language models:** they are still models with parameters, but at very large scale. Instead of two parameters, they may have billions; instead of a small training set, they are trained on very large text corpora; and instead of returning one simple numeric value, they perform inference over language.
- **Explain the key difference in input and output:** the input to an LLM is a sequence of words or tokens, and the immediate output is a prediction of one next token. More precisely, the model produces a probability distribution over known tokens, from which one token is selected.
- **Explain how this next-token prediction becomes a full answer:** the selected token is appended to the previous input sequence, the expanded sequence becomes the input for the next prediction, and this loop is repeated until a response is generated.
- **Explain why chat systems can refer to earlier parts of a conversation:** the model is not only given the latest user utterance, but a representation of the conversation so far, plus system instructions and other context made available to it.
- **Caveat to demonstrate:** because the model predicts a probability distribution and generation must select from plausible next tokens, the same prompt can produce different outputs across repeated runs, especially when sampling settings allow variation. This sensitizes participants to the fact that generative AI output is probabilistic rather than a fixed lookup result.
- **Emphasize the central learning point for Unit 1:** while LLMs are trained from data like other models, current LLM behaviour can also be controlled extensively through the input at inference time. Structured interaction therefore matters because the request itself becomes part of the condition under which the model predicts.

Recurring Agentic Coding Demonstration

Demonstrate how a vague coding request becomes a structured task brief in a real repository by making goal, repository context, constraints, files in scope, and expected output explicit. Add lightweight persistent project memory through a repo instruction file such as `AGENTS.md`, `CLAUDE.md`, or `GEMINI.md` so participants can see that good control begins before implementation. Compare a weak request with a structured brief and inspect the resulting plan, file targeting, and code response for differences in relevance, clarity, and usefulness.

Friday Activity Plan

Activity	Method	Tools and materials
From vague asking to structured interaction	Impulse lecture with worked examples	Slides, general chat interface
Comparing prompt structure variables	Guided pair experiment with prediction before execution	Chat interface, worksheet/template
Structured interaction across tools	Demonstration plus transfer exercise	Consensus or Perplexity, NotebookLM, GPT via Codex in Visual Studio Code
Challenge kickoff and first workflow draft	Individual or paired coaching; participants tailor the challenge to their own context and draft step-wise expected outputs	Selected AI tools, markdown working file

Detailed activity blocks

1

Activity From vague asking to structured interaction

Content focus. Contrast vague requests with well-formed requests by making goal, context, constraints, source expectations, and response form explicit; frame control at this stage as request design, not yet clarification-first dialogue or process design.

Method. Impulse lecture with worked examples

Tools and materials. Slides, general chat interface

2

Activity Comparing prompt structure variables

Content focus. Test how clearer task framing, explicit output structure, source expectations, and response constraints affect output quality; require participants to predict differences before running them.

Method. Guided pair experiment with prediction before execution

Tools and materials. Chat interface, worksheet/template

3

Activity Structured interaction across tools

Content focus. Show how the same request-design principles must be adapted across source discovery, grounded synthesis, and final document production; demonstrate deliberate transfer of outputs between tools without turning the session into workflow design.

Method. Demonstration plus transfer exercise

Tools and materials. Consensus or Perplexity, NotebookLM, GPT via Codex in Visual Studio Code

4

Activity Challenge kickoff and first workflow draft

Content focus. Help participants define topic, audience, document type, expected outputs per step, and first well-formed requests for discovery, synthesis, and production so the challenge begins from explicit request structure.

Method. Individual or paired coaching; participants tailor the challenge to their own context and draft step-wise expected outputs

Tools and materials. Selected AI tools, markdown working file

Experiment Focus

- **Experiment focus:** vary request structure
- **Manipulated variable, condition, or design decision:** Use the Friday activity table and challenge focus to define the comparison.
- **What remains constant across comparison:** Keep the underlying task or use case stable enough that observed differences can be interpreted credibly.

Observation Focus

- **Observation focus:** output fit and clarity
- **Evidence participants should capture:** prompts, outputs, traces, intermediate artefacts, screenshots, feedback, test results, or workflow records as appropriate for the unit.
- **Interpretation participants should be prepared to make:** Explain how the manipulated condition changed the observed outcome and what this reveals about control.

Challenge Kickoff Deliverables

Participants must leave Friday with:

- selected context-relevant task/use case
- baseline definition for comparison
- target condition or workflow idea
- first protocol draft covering objective/use case and intended output, manipulated design choices or variables, what remains constant across comparison, expected effects, and evaluation criteria
- first captured evidence, such as one run, one workflow sketch, one retrieval trace, one judge report, one responsibility protocol, or one capability map

The kickoff should already reflect the challenge-focus logic:

- **Difficulty:** Participants tend to rely on vague asking and lose control because the task is not made explicit enough within the request.
- **Approach:** Define well-formed requests in advance, predict where request structure should matter, run at least one structured-versus-weaker comparison, and deliberately transfer outputs from discovery to synthesis to final production.

- **Artefacts:** protocol document; final multi-page sectioned artefact; run log with prompts, transferred outputs, and observations; evidence excerpts or screenshots; concise interpretation; short expert or stakeholder feedback.

Teacher Setup

Mandatory:

- one or more examples that make the core distinction visible
- demonstration setup for the recurring agentic coding case, if used
- runnable environment(s) with low setup friction, plus fallback option
- protocol template and evidence-capture template

Configurable:

- exact tools/platforms
- **work mode:** individual, pair, or group
- scaffolding depth and peer-feedback format
- whether participants implement a workflow or tightly specify it when implementation is not feasible