CAS Human-AI Collaboration (Module 1)

Summary

Across nine iterative cycles, participants develop:

- the ability to predict AI behaviour
- the ability to design structured collaboration
- the ability to interpret results through evidence
- the ability to control outcomes across interaction, mechanism, data, system, and human context
- the ability to recognize and design for security-relevant trust boundaries
- the ability to adapt their approach as technologies and organizational conditions evolve

The course shifts focus from:

> *"What can AI do?"*

to:

> *"How do we design human-AI collaboration so that reliable, responsible outcomes emerge?"*

Overview and Structure

Module 1 develops **adaptive mastery in human-AI collaboration**. Participants learn to predict, design, and control AI behaviour by combining conceptual understanding, structured experimentation, inspectable artefacts, and coached transfer into their own professional contexts.

The module combines three recurring learning-content components:

- **Literacy perspectives**
  Interaction, Mechanism, Data, System, and Human-Context define the perspectives through which AI behaviour is observed and controlled.
- **Collaboration levels**
  The course moves from instruction and alignment toward co-construction, reflection, orchestration, and adaptive mastery.
- **Generative and agentic AI fundamentals**
  Each unit introduces a practical conceptual foundation of generative or agentic AI, including a demonstrable caveat that makes limits and failure modes observable.

Each Friday introduces and applies these components through conceptual input, guided experimentation, a recurring agentic coding demonstration, and challenge preparation. Each Saturday provides coaching and feedback on the corresponding challenge.

Across the nine challenges, participants build a cumulative evidence portfolio containing protocols, predictions, comparison outputs, observations, interpretations, and practical transfer rules. This portfolio provides the empirical foundation for the integrative project in Module 2.

How Levels, Literacies, and Fundamentals Work Together

The three learning components have different roles and should not be collapsed into one another.

Literacy perspectives explain what dimension of AI behaviour is being controlled:

- Interaction: request structure, exchange design, examples, clarification, and feedback
- Mechanism: tokens, attention, parameters, sampling, variability, and model behaviour

- Data: sources, grounding, embeddings, retrieval, factuality, and traceability
- System: workflows, handoffs, control points, tools, memory, and orchestration
- Human-Context: trust, responsibility, legitimacy, ethics, leadership, and social fit

Collaboration levels describe the developmental move in how participants work with AI:

- Level 0: asking for answers
- Level 1: instruction and structured request design
- Level 2: alignment and clarification before generation
- Level 3: co-construction and process design
- Level 4: reflection, critique, revision, and responsible judgment
- Level 5: orchestration across steps, tools, roles, and systems

Generative and agentic AI fundamentals provide the explanatory foundation:

- how generative AI models are trained, receive token-sequence input, and generate output through next-token prediction
- how context, tokens, sampling, embeddings, and grounding shape behaviour
- why critique, judging, responsibility, tools, and workflows need explicit design
- where limits, biases, instability, or risks remain demonstrable

The level defines the collaboration move. The literacy defines the experimental lens. The fundamental explains why the observed behaviour occurs.

Recurring Reference Application

A recurring instructor-side reference application in this module is **agentic work in a real development workspace**. An agentic coding environment may be used because it makes requests, context, intermediate artefacts, tool use, state, validation, and approval unusually visible.

This reference application is used where helpful to make collaboration moves concrete through a visible, inspectable workflow with artefacts such as specifications, plans, changes, tests, review outputs, workflow traces, reusable capabilities, resources, and approval gates.

It is not the main domain of the course and does not replace participants' own challenge contexts. The particular product, model, and infrastructure remain implementation choices. The reference case provides a stable demonstration that makes the weekly learning move easier to observe before participants transfer the same principle to their own professional or disciplinary use case.

The recurring agentic coding case is useful because it makes the main control dimensions of human-AI collaboration unusually visible in a single reference domain:

- Persistent context through durable instructions and reusable project information
- Specification before implementation through clarified scope, constraints, and acceptance criteria
- Evidence-bearing work through tests, checks, screenshots, and review findings
- Workflow over prompt through planning, coding, testing, review, approval, and integration
- Delegation and specialization through explicit roles and reviewer or debugger patterns
- Safety through permissions, restricted execution, checkpoints, and human approval
- Grounding through codebase files, tickets, documentation, screenshots, tool outputs, and external sources
- Security through source separation, least-privilege tool use, validation, audit traces, and approval checkpoints

Course Structure (Fridays and Challenges)

#	Friday Topic	Literacy Perspective	Collaboration Level	Fundamental	Recurring Agentic Coding Demonstration	Challenge
1	From Asking to Structuring: Designing AI Interaction	Interaction	Instruction	LLMs as next-token predictors; probabilistic output	Vague coding request versus structured task brief with project context	Research-Backed Professional Artefact
2	From Structuring to Alignment: Clarifying Before Generating	Interaction, secondarily Human-Context	Alignment	Context as the condition for prediction; context limits	Clarification-first coding specification versus direct-start implementation	Alignment by Clarification
3	Why Models Behave Differently: Parameters, Variability, and Control	Mechanism	Instruction with mechanistic control	Tokens, tokenisation, attention, sampling, and variability	Same coding task under different model or reasoning conditions	Parameter Control Experiment
4	From Opinion to Evidence: Grounding AI in Data and Sources	Data	Instruction with grounded evidence control	Embeddings, vectorisation, retrieval, grounding, and bias	Coding with minimal context versus grounded repo/API/test context	Grounded Generation Experiment
5	From Answers to Processes: Designing Human-AI Co-Construction	System, with Interaction	Co-construction	Decomposition, intermediate artefacts, roles, and process state	Single-pass coding versus phased specification, implementation, testing, and review	Co-Construction Workflow
6	Reflection Loops: Critique, Revision, and Better Reasoning	Interaction, secondarily Mechanism	Reflection	LLM-as-judge, conditions of satisfaction, and separate judging sessions	Baseline implementation plus critique, tests, review, and revision	Reflective Improvement Workflow
7	Leadership, Responsibility, and Legitimation in Human-AI Systems	Human-Context	Reflection with responsible co-construction	Fluent generation versus legitimate decision support; substantive oversight, contestability, and calibrated reliance	Consequential coding proposal with varied authority, evidence, approval, and escalation logic	Responsible Decision Logic Experiment
8	Designing Workflows: From Single Chat to Reliable AI Systems	System	Orchestration	Agentic AI: perception, reasoning, planning, tools, reusable capabilities, workflow state, and control points	Coding agent using persistent context, tools, resources, permissions, validation, and approvals	Workflow Control
9	Security and Trust Boundaries in Human-AI Collaboration	System, Data, and Human-Context, with Interaction as the attack surface	Orchestration with security-aware control	LLM security fundamentals: instruction/data boundaries, prompt injection, poisoned sources, permissions, and residual risk	Agentic coding or workflow case with untrusted source content, restricted permissions, validation, and approval checkpoints	Security Control Experiment

Relation to Module 2

Module 1 culminates in security-aware orchestration. The broader integrative work-practice redesign takes place in Module 2, where participants use selected evidence from their nine-unit portfolio to develop a thesis-like Human-AI Collaboration Blueprint and a personal theory of control.

The separate Module 2 overview describes this integrative project orientation.

Challenge Design Logic

Each unit challenge turns the Friday content into an experiment in the participant's own context.

Every challenge should make explicit:

- **Difficulty:** what is not yet controlled
- **Approach:** what condition, variable, or design choice is manipulated or compared
- **Artefacts:** what makes the outcome inspectable
- **Evidence:** what observations, traces, outputs, feedback, or measurements support the interpretation
- **Interpretation:** what the participant concludes about control, transfer, and limits

The challenges are not generic homework. They are the main vehicles through which participants develop and demonstrate predictive, evidence-based mastery.

Companion Design Documents

Compact unit-by-unit summaries and detailed teacher-facing delivery specifications are maintained as companion design documents. Module 1 remains the overview of the curriculum progression rather than the delivery script for individual Fridays and challenges.