



ARTIFICIAL INTELLIGENCE  
FINANCE INSTITUTE

## AI AGENTS IN FINANCE CERTIFICATE

**FEBRUARY 2026**  
OUTLINE COURSE



## COURSE DESCRIPTION

This intensive, hands-on course explores cutting-edge **Large-Language-Model (LLM) Agents** and their hybridisation with reinforcement learning in the financial domain. Students design, orchestrate, interpret and govern agentic systems for algorithmic trading, fraud detection, risk management and personalised advisory. Every session features substantial coding labs using production frameworks (OpenAI Responses SDK, AutoGen, CrewAI, LlamaIndex, LangChain) alongside emerging RL-tool-use libraries (ARTIST, ReCall, verl).

---

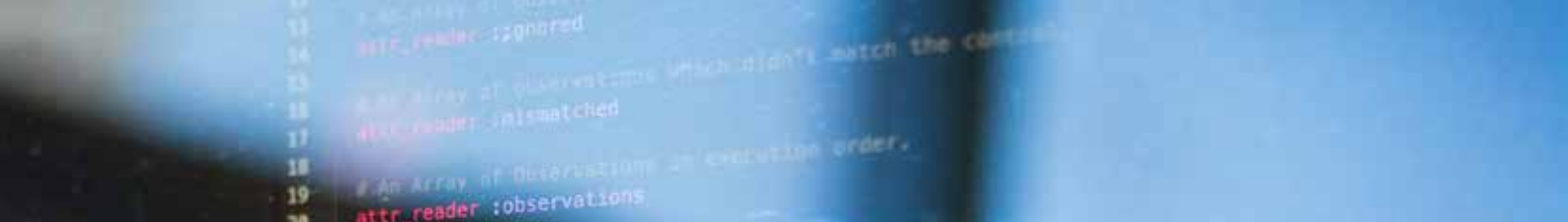
## PREREQUISITES

- > Working knowledge of Python programming.
  - > Introductory understanding of financial markets and instruments.
  - > Basic familiarity with machine-learning concepts; prior RL is helpful but not required.
- 

## LEARNING OUTCOMES

### **Upon completion students will be able to:**

- > Distinguish business workflows, AI-augmented workflows and autonomous AI agents.
- > Combine orchestration, retrieval and RL frameworks for finance.
- > Implement multi-agent crews that plan, code and self-correct using OpenAI Responses, AutoGen and CrewAI.
- > Build knowledge-integrated agents with Corrective-RAG and thought-tracing hooks.
- > Augment LLM agents with RL reward layers for tool selection, compliance and performance.
- > Evaluate agent safety with AgentBench tasks, tail-risk metrics and audit trails.



## COURSE OUTLINE (21 HOURS)

### **SESSION 1** Foundations - Workflows – AI Workflows – Agents (3 h)

- > Working knowledge of Python programming.
- > Introductory understanding of financial markets and instruments.
- > Basic familiarity with machine-learning concepts; prior RL is helpful but not required.

### **SESSION 2** Multi-Agent Architectures & Collaboration Patterns (3 h)

- > Crew roles, dynamic spawning, delegation and debate.
- > Communication schemas, reflection, voting.
- > Lab: three-agent investment committee parses SEC filings and issues a decision memo.

### **SESSION 3** Interpretability, Safety & Thought Tracing (3 h)

- > Anthropic circuit tracing; scratch-pad logging & sanitisation.
- > Governance: audit trails, pausable workflows.
- > Lab: instrument advice agent with circuit traces and auto-redaction.

### **SESSION 4** Tool-Using & Coding Agents (3 h)

- > Coder–Critic–Executor loops (Reflexion, Self-Refine).
- > Safe sandboxing and unit-test-driven repair.
- > MCP and advanced tools
- > Lab: repair-bot iteratively fixes a faulty VaR calculator.

## SESSION 5 Knowledge-Integrated Agents & RAG 2.0 (3 h)

- > Corrective-RAG (CRAG), Introspective Agents, structured retrieval.
- > Stateful agents with Letta
- > Streaming pipelines, vector-DB ops, compliance logging.
- > **Lab:** streaming RAG agent ingests live 10-Q filings and flags covenant breaches.

## SESSION 6 LLM Agents + RL Layers in Production Finance (3 h)

- > **RL Adds Value:** RLHF/RLAIF and policy-level RL (ARTIST) for tool-selection and compliance.
- > Turn-level credit assignment; ReCall & Agent-R1 for end-to-end tool-calling RL.
- > **Hybrid market simulators:** TradingAgents crews, FinRL-Meta; LLM-only vs PPO trader comparison.
- > **Evaluation:** AgentBench finance tasks, CRAG-based factual rewards, Self-Taught Evaluator.
- > **Lab:** four-agent prop-desk crew (planner, analyst, coder, risk officer) deployed in FinRL-Meta equities env-targets: Sharpe, CVaR, compliance.

## SESSION 7 Evaluation, Governance & Deployment (3 h)

- > **Benchmarks:** AgentBench, SWE-agent regression.
- > **Risk metrics:** CVaR, stress scenarios, fail-rate.
- > **Deployment pipelines:** blue-green, guard-rails, kill-switches, EU AI Act mapping.
- > **Capstone:** monitoring dashboards and audit procedures.

```
19
20 att_reader : observations
21
22 # Internal: Create a new result.
23 #
24 # experiment - the Experiment this result is for
25 # observations - an Array of Observations, in ascending order
26 # control - the control Observation
```

## ASSESSMENT METHODS

- > **Practical Labs (60%)** – graded coding exercises.
- > **Final Project (30%)** – finance-sector agent meeting safety & performance targets.
- > **Participation (10%)** – engagement and peer reviews.

## RECOMMENDED RESOURCES

- > **Frameworks:** OpenAI Responses, AutoGen, CrewAI, LlamaIndex, LangChain.
- > **RL & Tool-use:** ARTIST, ReCall, verl.
- > **Benchmarks/Simulators:** AgentBench, TradingAgents, FinRL-Meta, SWE-agent.
- > **Key Papers:** Tree-of-Thoughts, Algorithm-of-Thoughts, CRITIC, Corrective-RAG, Turn-Level Credit Assignment, Self-Taught Evaluator.
- > **Interpretability:** Anthropic Circuits, Thought-Tracing Toolkit.

## INSTRUCTORS

- > Nickole Koenigstein
- > Miquel Noguer i Alonso
- > David Pacheco Aznar
- > Miquel Trafi Ruiz

## COURSE LOGISTICS

- > **Course Duration** 21 Hours (7 lectures & 7 Office Hours)
- > **Start Date** Monday, 2th February 2026
- > **Course Fee** Standard Price: 2,000 USD
- > **Discounts** **Super Early Bird:** 20% discount (1,600 USD) – Available until 28th November 2025  
**Early Bird:** 10% discount (1,800 USD) – Available until 9th January 2026

```

1 # Scientist::Result
2
3 # We have to compare observations
4 with reader candidates
5
6 # The wrong observation is when the test are compared
7 with reader control
8
9 # Public:
10
11 # Public: the experiment
12 with reader experiment
13
14 # Public: the observation's will read the control, but when compared
15 with reader ignored
16
17 # Public: the observation's will read the control
18 with reader mismatched
19
20 # An Array of observations in execution order
21 with reader observations
22
23 # Internal: Create a new result
24
25 # experiment - the Experiment this result is for
26 # observations - an Array of Observations, in execution order
27 # control - the control Observation
28
29 def initialize(experiment, observations = [], control = nil)
30   @experiment = experiment
31   @observations = observations
32   @control = control
33   @candidates = observations - [control]
34   evaluate_candidates
35
36   freeze
37 end
38
39 # Public: the experiment's context
40 def context
41   experiment.context
42 end
43
44 # Public: the name of the experiment
45 def experiment_name
46   experiment.name

```



ARTIFICIAL INTELLIGENCE  
FINANCE INSTITUTE