

AI Agents and Tool Use

Tool calling, MCP servers, the agent loop, and the permissions around it. You build agents that finish the task instead of looping forever, and you can show exactly what they did.

Advanced

4 phases

22 sessions

33 contact hours

Syllabus

4 phases, 22 sessions. Each phase ends in something you have built.

1

The Loop, By Hand

Understand what an agent is before a framework hides it from you.

5 sessions

\$108 alone

1.1 The Loop Is Ten Lines

Write the while loop yourself: model call, tool call, append the result, check the stop condition.

[FREE PREVIEW](#)

1.2 Tool Schemas That Work

Naming, descriptions, enums, and the JSON Schema fields a model actually pays attention to.

1.3 Errors the Model Can Read

Bad arguments, hallucinated tool names, and error text written for a model instead of a human.

1.4 Plan or React

ReAct, upfront plans, replanning, and the tasks where planning costs more than it saves.

1.5 Budgets and Stop Conditions

Turn caps, wall clock, token and dollar ceilings enforced in code, not asked for in the prompt.

By the end: **A framework-free agent loop with a turn cap, a dollar cap, and a trace you can read line by line.**

2

Tools and the MCP Layer

Connect an agent to real systems over MCP without trusting whoever wrote the tool.

6 sessions

\$108 alone

2.1 MCP, Concretely

Tools, resources, streamable HTTP, and what the 2026-07-28 revision moved into extensions.

2.2 Writing a Server

Build a server with typed input and output schemas, then drive it from two different clients.

2.3 Auth Without Confused Deputies

OAuth 2.1, protected resource metadata, and the audience check that stops token passthrough.

2.4 Tool Poisoning and Rug Pulls

Instructions hidden in tool descriptions, line jumping at tools/list, and pinning what you approved.

2.5 Schema Drift

Hash the whole tool surface, diff it on every deploy, and catch the silent breaking change.

2.6 Too Many Tools

150,000 tokens of definitions down to 2,000: progressive disclosure and code execution over MCP.

By the end: **An MCP server of your own with OAuth, plus a client whose tool surface is hashed and pinned.**

3

State That Survives

Give the agent state that outlives the process and memory that earns its tokens.

5 sessions

\$108 alone

3.1 Why a Graph at All

create_agent, StateGraph, and the point where a hand-rolled loop stops being worth keeping.

3.2 Checkpointers and Threads

State reducers, a Postgres checkpointer, and a thread id you own rather than one you invent.

3.3 Long-Term Memory

Namespaced writes to a store, scoped retrieval, and a TTL so stale facts stop coming back.

3.4 Context Rot

Degradation measured at 100k tokens, schema-driven compaction, and what the second summary dropped.

3.5 Durable Resume

Checkpoints are not durable execution: idempotency keys, at-least-once steps, replay after kill -9.

By the end: **A long-running agent killed mid-run and resumed without repeating a single side effect.**

4

Blast Radius and Bills

Run the thing somewhere it can be watched, stopped, and paid for.

6 sessions

\$108 alone

4.1 Sandboxing for Real

Shared-kernel containers against Firecracker microVMs, and egress rules that matter more than both.

4.2 Permissions and Capabilities

Read-only by default, scoped credentials, and the Replit agent that dropped a production database.

4.3 Approval a Human Can Give

interrupt(), a diff a reviewer can actually judge, and what happens when nobody clicks for a week.

4.4 Tracing the Whole Run

OpenTelemetry GenAI spans, run and turn hierarchy, and finding the step that actually broke.

4.5 The Runaway Drill

Build a two-agent clarification loop on purpose, watch the spend curve, then stop it server side.

4.6 When Not to Multi-Agent

Supervisor and handoff patterns, a 15x token bill, and the equal-budget single-agent baseline.

By the end: **A sandboxed agent with approval gates, full traces, and a spend cap proven by a runaway drill.**

Tools you will use

LangGraph 1.2

LangChain 1.3

Model Context Protocol 2026-07-28

MCP Python SDK

Pydantic AI 2.x

LangSmith

Langfuse

OpenTelemetry GenAI conventions

E2B / Modal sandboxes

Temporal

What you will build

1 The Bounded Loop

Hand-written agent loop, hard caps, a readable trace

2 Your Own MCP Server

OAuth, pinned schemas, and a rug-pull test that fails loudly

3 The Agent That Survives

Killed mid-run, resumed, sandboxed, traced, under budget

Registration

Phase 1: The Loop, By Hand	\$108
Phase 2: Tools and the MCP Layer	\$108
Phase 3: State That Survives	\$108
Phase 4: Blast Radius and Bills	\$108
Whole track, all 4 phases	\$360

Buying every phase separately costs \$432, so the whole track saves \$72.

No payment is taken online. Registering creates an order and payment instructions are sent with it. Access opens once payment is confirmed.