COMPANION TO THE AI & AGENTIC AI ENGINEER PROGRAMME

Detailed Syllabus

Session-by-session topic breakdown for the AI & Agentic AI Engineer Programme - every day of all sixteen weeks, mapped to what you will actually learn.

80	16	9	1.5 hrs
sessions total	weeks	phases	per session

HOW TO READ THIS DOCUMENT

Each session below is numbered D01 through D80 and grouped by week within its phase, matching the roadmap in the programme brochure - from Python and LLM foundations through RAG, agent frameworks, multi-agent protocols, evaluation, deployment and the capstone.

PHASE 01 Python & LLM Foundations — WEEKS 1-2

WEEK 1

D01	**Programme orientation & the AI engineer's workflow** —From prototype to production; the toolchain this programme covers
D02	**Python for AI systems** —Advanced functions, decorators, async basics
D03	**Working with LLM APIs** —OpenAI and Anthropic API fundamentals
D04	**Understanding LLM behaviour** —Tokens, context windows, temperature, sampling
D05	**Building a basic LLM-backed script** —Chaining a simple prompt-response workflow

WEEK 2

D06	**FastAPI fundamentals** —Building a simple REST API
D07	**Serving an LLM call through an API endpoint** —Wrapping a model call as a service
D08	**Open-source & self-hosted models** —When and why to use them instead of a hosted API
D09	**Error handling & retries for LLM calls** —Rate limits, timeouts, fallback strategies
D10	**Milestone check-in: a small LLM-backed API** —Building the pieces from week 1-2 into one working service

PHASE 02 Prompt Engineering & Structured Outputs — WEEK 3

WEEK 3

D11	**Advanced prompting patterns** —Chain-of-thought, few-shot examples
D12	**Function calling** —Defining tools an LLM can call
D13	**Structured outputs** —JSON schema enforcement, Pydantic models
D14	**Prompt caching & cost/latency basics** —Token economics for production systems
D15	**Milestone project: Prompt Engineering & Structured Output** —A reliable function-calling pipeline with clean output

PHASE 03 RAG & Retrieval Systems — WEEKS 4-5

WEEK 4

D16	**Embeddings fundamentals** —How text becomes vectors
D17	**Vector databases** —Pinecone, Qdrant and Weaviate basics
D18	**Building a basic retrieval pipeline** —Indexing and querying documents
D19	**Chunking strategies** —Splitting documents for effective retrieval
D20	**Hybrid retrieval** —Combining keyword and semantic search

WEEK 5

D21	**Reranking** —Improving retrieval precision
D22	**RAG architecture end to end** —Connecting retrieval to generation
D23	**Citations & source grounding** —Making an answer traceable back to its source
D24	**RAG practice** —Building a grounded Q&A system
D25	**Milestone project: RAG Retrieval System** —A grounded retrieval pipeline over a real document set

PHASE 04 Agent Frameworks & Tool Use — WEEKS 6-8

WEEK 6

D26	**Introduction to LangChain** —Chains, prompts, output parsers
D27	**LangChain tools & tool calling** —Giving an LLM the ability to act
D28	**Introduction to LangGraph** —State machines for agents
D29	**Building an agent's reasoning loop** —Plan, act, observe
D30	**Agent memory** —Short-term vs long-term memory

PHASE 04 Agent Frameworks & Tool Use — WEEKS 6-8

WEEK 7

D31	**LlamaIndex for retrieval-heavy agents** —When to reach for it over LangChain
D32	**Multi-step tool use** —Chaining multiple tool calls together
D33	**Error recovery in agents** —Handling failed tool calls gracefully
D34	**Agent state management** —Tracking progress across steps
D35	**Building a research agent** —Practice with a multi-step, tool-using agent

WEEK 8

D36	**Agent planning strategies** —ReAct and plan-and-execute patterns
D37	**Human-in-the-loop patterns** —Approval steps inside an agent workflow
D38	**Testing agents** —Unit testing tool-calling logic
D39	**Agent practice** —Debugging a multi-step agent
D40	**Milestone project: Single-Agent Tool-Use** —An agent that plans, calls tools, and completes a task

PHASE 05 Multi-Agent Systems & Protocols — WEEKS 9-10

WEEK 9

D41	**Multi-agent system design** —When to use multiple agents instead of one
D42	**Agent communication patterns** —Message passing and shared state
D43	**Introduction to the Model Context Protocol (MCP)** —A standard way to expose tools to agents
D44	**Building an MCP server** —Exposing your own tools over MCP
D45	**Connecting agents to MCP servers** —Wiring an agent up to external tools

PHASE 05 Multi-Agent Systems & Protocols — WEEKS 9-10

WEEK 10

D46	**Introduction to Agent2Agent (A2A) communication** —How agents talk to other agents
D47	**Orchestrator-worker agent patterns** —One agent directing several others
D48	**Coordinating agents on a shared goal** —Avoiding conflicting actions between agents
D49	**Multi-agent practice** —Debugging coordination issues
D50	**Milestone project: Multi-Agent Orchestration** —A small team of agents coordinating via MCP

PHASE 06 Evaluation, Observability & Safety — WEEKS 11-12

WEEK 11

D51	**Why agent evaluation is different** —Evaluating a process, not just an output
D52	**Building an eval suite** —Accuracy, relevance, task completion
D53	**RAGAS-style evaluation for retrieval quality** —Scoring a RAG pipeline properly
D54	**LLM-as-judge evaluation patterns** —Using a model to grade another model's output
D55	**Tracking cost & latency metrics** —Keeping a production system affordable and fast

WEEK 12

D56	**Observability tooling** —Tracing an agent's run step by step
D57	**Prompt-injection defence** —Protecting an agent from adversarial input
D58	**Guardrails** —Input and output filtering
D59	**Human-in-the-loop checkpoints** —Escalation points for high-risk actions
D60	**Milestone project: Agent Evaluation & Observability** —An eval suite scoring accuracy, latency and cost

PHASE 07 Deployment & AgentOps — WEEKS 13-14

WEEK 13

D61	**The AgentOps lifecycle** —From prototype to a production service
D62	**Packaging an agent as a service** —Structuring the code for deployment
D63	**Sandboxed execution** —Running agent-generated code safely
D64	**Introduction to Docker for AI services** —Containers vs local environments
D65	**Containerising an agent API** —Writing a Dockerfile, building an image

WEEK 14

D66	**Deploying a containerised agent** —Getting the container running as a live service
D67	**Monitoring agents in production** —Logging and alerting
D68	**Cost & latency optimisation** —Caching and model selection strategies
D69	**Handling agent failures in production** —Fallbacks and circuit breakers
D70	**Milestone project: Deployed Agent With Guardrails** —A live, monitored agent API

PHASE 08 AI Engineering Communication — WEEK 15

WEEK 15

D71	**Documenting an agentic system** —Architecture diagrams and READMEs
D72	**Writing a model/agent card** —Capabilities, limitations and intended use
D73	**Presenting to technical stakeholders** —Explaining architecture and tradeoffs
D74	**Presenting to non-technical stakeholders** —Framing an agent in terms of business value
D75	**Running a live technical demo** —Practice presenting a working system

PHASE 09 Capstone Project — WEEK 16

WEEK 16

D76	**Capstone kickoff** —Choosing a track and planning the approach
D77	**Work session: building the core agent** —Implementing the main capstone system
D78	**Work session: evaluation & guardrails** —Testing and safety-checking the agent
D79	**Work session: deployment & monitoring setup** —Getting the capstone live
D80	**Final capstone presentations** —Presenting the completed capstone; programme wrap-up