



Mastering Generative & Agentic AI with Python

10-WEEK INTENSIVE PROGRAM

From Python Foundations to Azure AI Foundry and production-grade Agentic Systems — this intensive program takes software engineers and ML practitioners through every layer of the modern AI stack. In 10 focused weeks, you'll build real systems, not just prototypes.

Foundation

Weeks 1–4: Python, ML, Deep Learning

Generative AI

Weeks 5–7: LLMs, RAG, Azure, Safety

Agentic Systems

Weeks 8–10: Agents, Orchestration, Deploy

+91 91 5239 5239 | +91 9059 16 9059

Weeks 1–2: Environment & Python Foundations

CHAPTER 1 • FOUNDATIONS

Week 1 — Professional Python Environment

Build a production-ready development baseline from day one. You'll configure Python 3.11+, set up VS Code and Jupyter, enforce code quality with **black**, **ruff**, and **mypy**, and wire up a CI pipeline using GitHub Actions or Azure Pipelines.

- Git workflows & structured repo layout
- pytest + coverage enforcement
- CI: lint + test + format on every push

Week 2 — Python Fluency for AI Systems

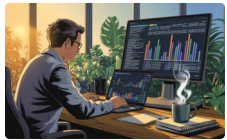
Go beyond syntax. This week targets the Python patterns that power backend and AI workflows: **async programming** with asyncio, Pydantic for typed config, OOP design with dataclasses, and structured logging.

- Generators, closures, and type hints
- Async CLI log parser lab
- Typed config loader with Pydantic

- ✓ Outcome: A fully linted, tested, CI-enforced Python environment — the foundation every production AI system depends on.

Weeks 3–4: ML Foundations & Deep Learning

CHAPTER 1 · FOUNDATIONS



Week 3 — ML Workflow & Evaluation

Master the full ML lifecycle using **pandas**, **scikit-learn**, and **MLflow**. Build regression and classification pipelines, apply rigorous model validation, track experiments with metrics and artifacts.

- Scikit-learn pipeline architecture
- Model metrics & cross-validation
- MLflow experiment tracking lab



Week 4 — PyTorch & Reproducibility

Understand training loop anatomy with

PyTorch. Train a multilayer perceptron on tabular data, log metrics and artifacts, and learn how to package ML systems for reproducibility across environments.

PyTorch fundamentals & autograd

Reproducible ML packaging

Artifact logging & model evaluation

Week 5: Generative AI Fundamentals

CHAPTER 2 · GENERATIVE AI

Shift from classical ML to the transformer era. This week demystifies how large language models work, how to engineer effective prompts, and how to evaluate outputs with rigor rather than intuition.



LLM Fundamentals

Attention mechanisms, context windows, and how transformer models generate text token by token.



Tokenization & Embeddings

Understand how text becomes vectors. Explore embedding spaces and semantic similarity at a practical level.



Prompt Engineering

System prompts, few-shot examples, chain-of-thought — practical patterns for controlling LLM behavior reliably.



Evaluation Basics

Groundedness, faithfulness, and relevance. Build a test dataset and evaluate prompt variants systematically.

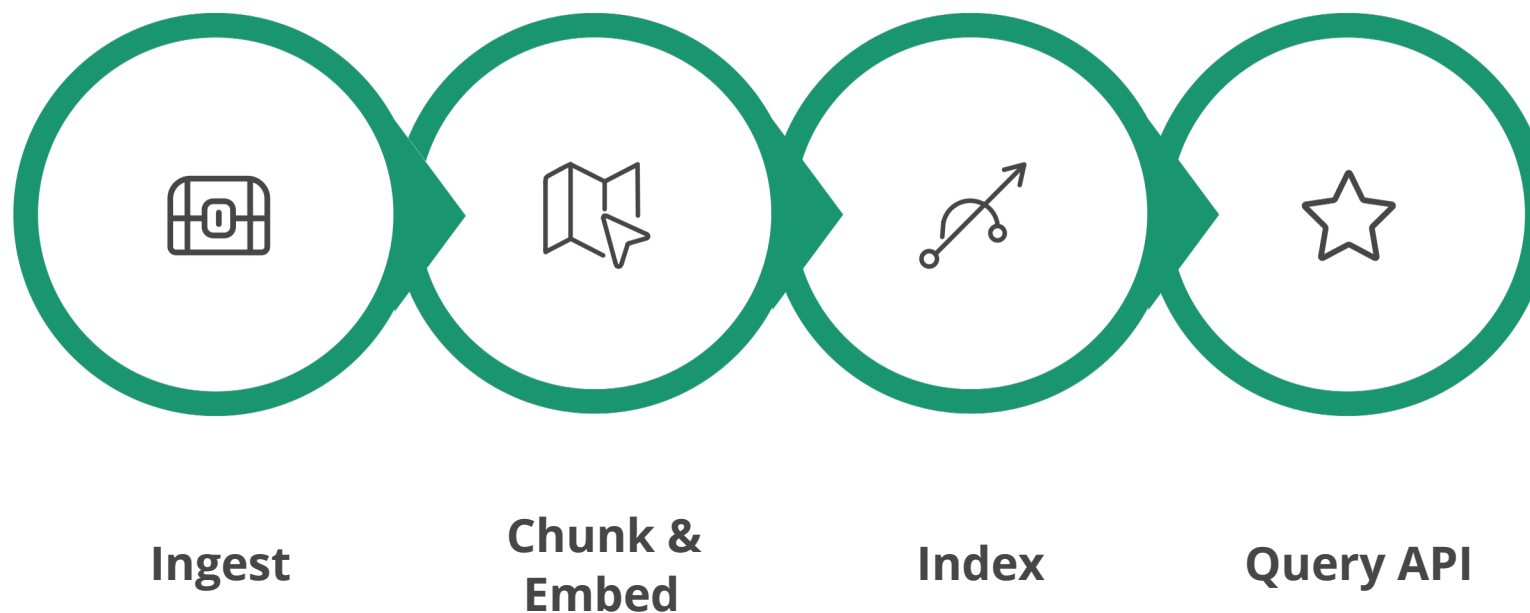


Lab: Prompt experiments notebook + systematic evaluation against a curated test dataset.

Week 6: Azure AI Foundry & RAG Systems

CHAPTER 2 · GENERATIVE AI

Build a production-grade **Retrieval-Augmented Generation** system on Azure end-to-end. From blob ingestion to vector search to a live FastAPI endpoint — this is where theory meets deployed infrastructure.



Core Topics

- Azure AI Foundry project setup
- Azure OpenAI deployment configuration
- Chunking strategies & metadata design
- Embeddings + Azure AI Search indexing

Hands-On Lab

- End-to-end RAG API with FastAPI
- Blob ingestion pipeline
- Vector search integration
- Query latency benchmarking

Week 7: Safety, Guardrails & Optimization

CHAPTER 2 · GENERATIVE AI

Production GenAI systems must be safe, cost-aware, and performant. This week adds the critical safety and optimization layer that separates prototypes from deployable systems.

Azure AI Content Safety

Integrate Azure's content safety API to screen inputs and outputs. Add a moderation layer that blocks harmful content before it reaches your application or users.

Structured Outputs & JSON Mode

Force reliable, schema-validated responses using JSON mode and function-calling patterns. Essential for downstream system integrations.

Streaming Responses

Implement streaming endpoints with FastAPI and the Azure OpenAI SDK to dramatically improve perceived latency and user experience.

Cost & Latency Optimization

Token budgeting, caching strategies, and model tier selection. Learn to instrument and optimize your GenAI system for production economics.



▲ Safety is not optional. Every production GenAI system requires content moderation, output validation, and cost guardrails before deployment.

Week 8: Agentic AI Architecture

CHAPTER 3 · AGENTIC SYSTEMS

Move beyond single-turn LLM calls. Agents plan, use tools, reflect on results, and maintain memory across multi-step tasks. This week covers the architecture patterns that make agents reliable and composable.

1

Plan

Break task into sub-goals using LLM reasoning

2

Act

Call tools — web search, PDFs, APIs — with schema validation

3

Reflect

Evaluate results, handle errors, update the plan

4

Remember

Short-term context + long-term memory with vector store

LangChain Agents in Practice

Build a **research assistant agent** that integrates web search and PDF parsing tools. Implement LangChain's agent executor with proper tool schemas, error handling, and memory persistence.

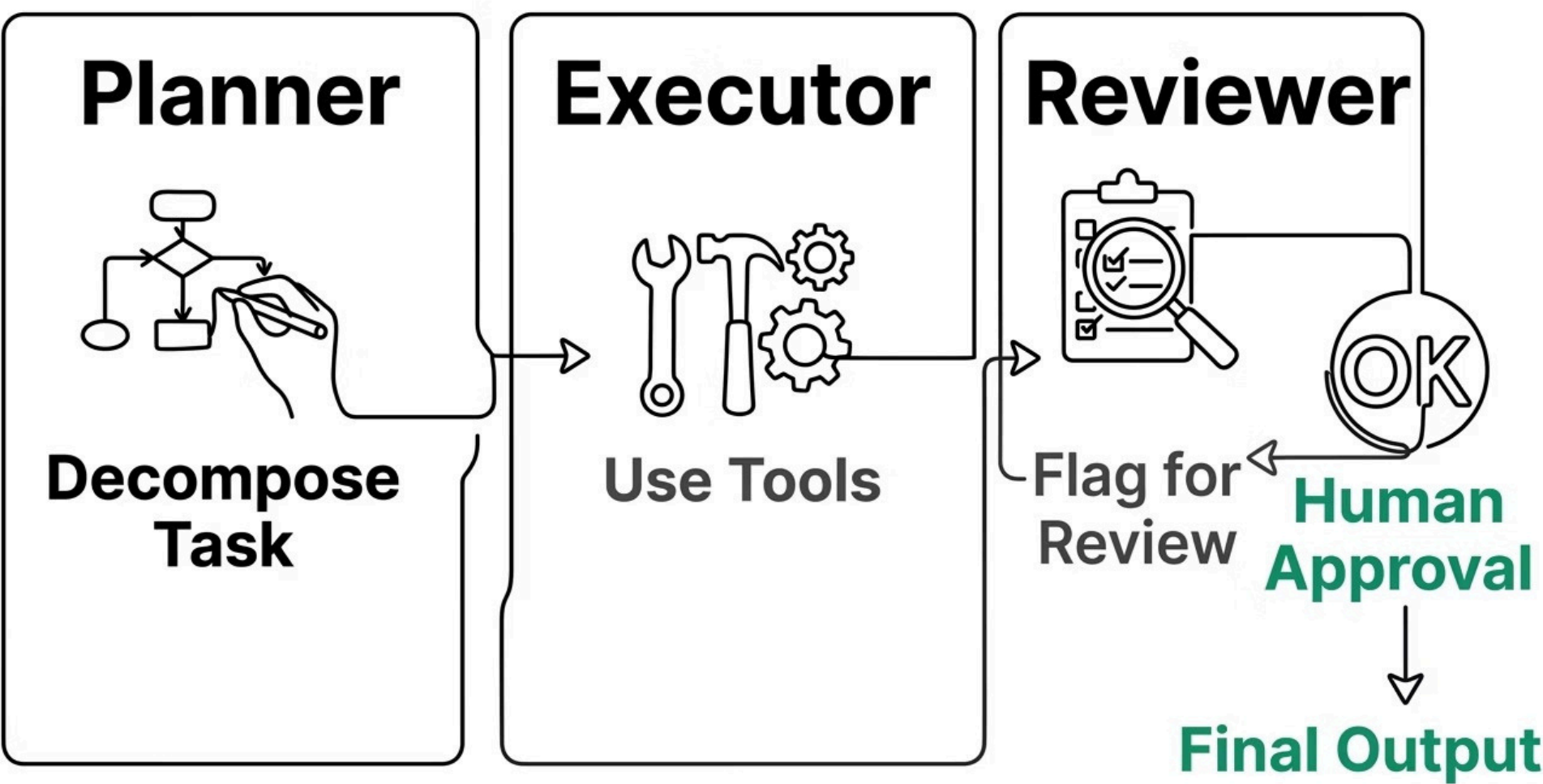
Lab Deliverable

A working research assistant agent with at least two tools, short-term conversation memory, and structured output logging.

Week 9: Multi-Agent Systems & Orchestration

CHAPTER 3 · AGENTIC SYSTEMS

Scale from single agents to coordinated multi-agent pipelines. Use **AutoGen** and **Azure Prompt Flow** to build orchestrated workflows with human approval gates, tracing, and replay capabilities.



AutoGen Flows

Multi-agent message passing with role separation and turn-taking control.

Approval Gates

Human-in-the-loop interrupts for sensitive actions or low-confidence decisions.

Prompt Flow

Azure orchestration for versioned, observable, multi-step agent pipelines.

Trace & Replay

Full execution logging with structured trace replay for debugging and audit.

Week 10: Deployment & Capstone

CHAPTER 3 · AGENTIC SYSTEMS

The final week brings everything together. Deploy a production-grade RAG or Agentic system to **Azure Container Apps**, secure it with Managed Identity and Key Vault, instrument observability, and present a complete architecture document.

Deployment Stack

- FastAPI containerized with Docker
- Azure Container Apps with auto-scaling
- Managed Identity — no secrets in code
- Azure Key Vault for credential management

Capstone Deliverables

- Deployed AI system (RAG or Agentic)
- Architecture documentation
- CI/CD pipeline with quality gates
- Cost model & observability dashboard

10

Weeks of Depth

Covering the full stack from environment setup to production deployment

5

Azure Services

OpenAI, AI Search, Container Apps, Key Vault, Content Safety

8+

Hands-On Labs

Every week ends with a real, deployable project artifact

Tech Stack, Assessment & Readiness

Every tool in this program is production-grade. The assessment rubric mirrors what engineering teams evaluate in code reviews, ML deployments, and system design interviews.

Core Tech Stack

Category	Tools
Runtime	Python 3.11+, FastAPI, asyncio, uvicorn
AI / GenAI	Azure OpenAI SDK, LangChain, LlamaIndex, FAISS
Azure SDKs	azure-identity, azure-search-documents, azure-storage-blob
Quality	black, ruff, mypy, pytest, coverage
Infra	Docker, Azure Container Apps, GitHub Actions

Assessment Rubric

- **Code quality & test coverage**
- **Model evaluation rigor**
- **Safety & guardrails implementation**
- **Observability & cost awareness**
- **Production readiness & documentation**



🚀 Graduates leave with a deployed AI system, a documented architecture, and a GitHub portfolio that demonstrates production-grade engineering — not just notebook experiments.