Faysal Aziz

AI-103 Accelerator

Building AI Apps and Agents on Azure - 3 days

FA-0080

Agentic & Generative AI · DevOps, Cloud & Infrastructure

Introduction

Microsoft's new AI-103 direction matters because it reflects a real shift in how AI systems are being built: away from isolated service demos and toward integrated applications, agent workflows, model evaluation, multimodal processing, and production deployment on Microsoft Foundry and Azure. Microsoft Learn lists **AI-103T00: Develop AI apps and agents on Azure** as a new intermediate course for AI engineers and developers, officially available from **March 31, 2026**, with a standard duration of **4 days**. Its stated focus is building AI-infused applications with Microsoft Foundry, including generative AI apps, AI agents, tool and knowledge connections, multimodal capabilities, and complex-content understanding.

This 3-day outline is designed as a compressed, industry-style version of that curriculum rather than an academic survey. It also accounts for Microsoft's transition away from **AI-102**. Microsoft lists **AI-102T00** as retiring on **April 30, 2026**, with **AI-103T00** named as its replacement course; separately, Microsoft lists the **AI-102 certification exam** as retiring on **June 30, 2026**. Microsoft's March 2026 certification announcement further explains that AI-103 aligns to newer generative and agentic architectures, with beta timing in **April 2026** and general availability expected in **June 2026**. Community replies from Microsoft Learn experts also indicate that AI-102 and AI-103 are **not equivalent**, and there is no automatic transition from the old certification to the new one.

The instructor for this program brings **over 30 years of industry experience** and will teach the material through current, employer-relevant practice: what teams are actually building now with Azure AI Foundry, model catalogs, evaluation workflows, agent frameworks, orchestration, and multimodal pipelines, rather than treating AI as a purely theoretical subject.

Learning outcomes

- Plan and structure AI application development on Azure using Microsoft Foundry and related Azure services.
- Build generative AI applications with deployed models, SDKs, prompts, and application integration patterns.

- Develop AI agents with Microsoft Foundry Agent Service and Microsoft Agent Framework.
- Extend agents with tools, functions, knowledge connections, and workflow orchestration patterns.
- Evaluate generative AI behavior and apply responsible AI practices in Azure AI Foundry.
- Design solutions that incorporate multimodal content and complex-content understanding.
- Recognize where legacy AI-102-era skills still matter and where the new AI-103 emphasis changes architecture and delivery choices.

Prerequisites

- Working knowledge of **Python**. Microsoft positions AI-103 for developers already familiar with Python and APIs/SDKs on Azure.
- Basic familiarity with **Azure** and the **Azure portal**.
- Comfort using **APIs, SDKs, and application integration patterns**.
- Foundational understanding of **generative AI concepts**.
- Helpful but not mandatory: prior exposure to AI-102 topics such as Azure AI services, language, vision, or information extraction.

Training outline

1. Course positioning and Azure AI landscape

The Microsoft AI certification and course transition

1. What AI-102 covered
2. Why Microsoft introduced AI-103
3. Courseware retirement vs exam retirement
4. Implications for teams, learners, and certification paths

The modern Azure AI application stack

1. Microsoft Foundry and Azure AI services
2. Foundry projects, resources, and environments
3. Models, agents, tools, and knowledge as first-class building blocks

Delivery mindset for production AI

1. From prototype to deployable application
2. Developer responsibilities in AI solution lifecycles
3. Responsible AI as an engineering concern, not a compliance afterthought

2. Planning and preparing to develop AI solutions on Azure

Solution architecture fundamentals

1. Selecting the right Azure AI components
2. Matching use cases to models, tools, and orchestration styles
3. Choosing between direct app patterns and agent patterns

Environment and resource preparation

1. Azure subscriptions, resource groups, and access setup
2. Foundry workspaces, projects, and resource organization
3. Model access, deployment readiness, and configuration baselines

Security and governance foundations

1. Identity, secrets, and access boundaries
2. Development vs test vs production considerations
3. Governance checkpoints for AI workloads

Application lifecycle planning

1. Prototyping, iteration, evaluation, and release flow
2. Traceability and reproducibility expectations
3. Operational readiness criteria

3. Building generative AI applications in Azure

Generative AI application fundamentals

1. Core prompt-response application patterns
2. System instructions, user prompts, and output shaping
3. Stateless vs stateful interaction design

Working with models in Azure AI Foundry

1. Model catalog concepts
2. Model selection criteria
3. Deployment configuration and inference access

Developing with the Azure AI Foundry SDK

1. SDK structure and project setup
2. Connecting applications to deployed models
3. Managing requests, responses, and runtime settings

Prompt and interaction design

1. Instruction design patterns
2. Output constraints and formatting control

3. Grounding strategies for better reliability

Integrating generative features into applications

1. Service-layer integration
2. Web and API integration patterns
3. Handling latency, retries, and failure modes

Responsible generative AI implementation

1. Safety-by-design principles
2. Content risk awareness
3. Human oversight and escalation design

4. Evaluating and improving generative AI performance

Why evaluation matters

1. Functional quality vs model quality
2. Reliability, relevance, and consistency
3. Evaluation in iterative delivery cycles

Evaluation workflows in Azure AI Foundry

1. Baseline creation
2. Test dataset setup
3. Comparative evaluation approaches

Assessment dimensions

1. Accuracy and helpfulness
2. Groundedness and hallucination risk
3. Safety and policy alignment
4. Output consistency and task completion

Tuning and improvement loops

1. Prompt refinement
2. Model substitution and comparison
3. Tool-use adjustments
4. Knowledge-source refinement

Release readiness

1. Acceptance thresholds
2. Regression checks
3. Monitoring expectations after deployment

5. AI agent fundamentals on Azure

What makes an AI agent different from a standard AI app

1. Goal-directed behavior
2. Tool use and action-taking
3. Multi-step reasoning and workflow execution

Agent use-case identification

1. When an agent is appropriate
2. When a simple app pattern is better
3. Boundaries of autonomy in enterprise systems

Microsoft Foundry Agent Service concepts

1. Agent architecture
2. Instructions, memory, tools, and knowledge connections
3. Runtime interaction model

Microsoft Agent Framework concepts

1. SDK role in application development
2. Agent composition principles
3. Framework-level orchestration patterns

6. Developing AI agents with Microsoft Foundry and Visual Studio Code

Agent development workflow

1. Project scaffolding
2. Agent configuration
3. Development toolchain setup

Building the first agent

1. Defining purpose and behavior
2. Configuring prompts and instructions
3. Establishing execution boundaries

Extending agent capabilities

1. Tool registration
2. Function calling patterns
3. Connecting business logic and APIs

Testing and debugging agents

1. Playground-based testing

2. Prompt and tool debugging
3. Behavioral validation across scenarios

Deploying and integrating agents

1. Deployment packaging
2. Application integration
3. Operational verification

7. Developing agents with Microsoft Agent Framework

Framework architecture

1. Core abstractions
2. Agent definitions and execution flow
3. Tool and connector integration

Agent creation patterns

1. Single-purpose agents
2. Task-oriented agents
3. Knowledge-enabled agents

Tool augmentation

1. Calling external functions
2. Structuring actions and responses
3. Managing tool-result feedback loops

Enterprise integration

1. Linking agents to internal services
2. Designing controlled action paths
3. Reliability and maintainability considerations

8. Orchestrating multi-agent solutions

Multi-agent architecture fundamentals

1. Coordinator and specialist patterns
2. Delegation and handoff
3. Conversation and state boundaries

Orchestration strategies

1. Sequential workflows
2. Parallel agent collaboration
3. Escalation and fallback paths

Workflow composition in Microsoft Foundry

1. Agent-driven workflows
2. Task decomposition
3. Result aggregation

Governance for multi-agent systems

1. Control points
2. Observability needs
3. Risk management in distributed reasoning flows

9. Knowledge connections and retrieval-enabled applications

Why grounding matters

1. Enterprise knowledge challenges
2. Reducing hallucination in production scenarios
3. Retrieval as part of application design

Knowledge-source integration patterns

1. Document collections
2. Structured and semi-structured content
3. External enterprise knowledge endpoints

Designing retrieval-enhanced responses

1. Query formulation
2. Context injection
3. Citation and answer traceability patterns

Tool vs knowledge decisions

1. When an agent should retrieve
2. When an agent should act
3. Combining retrieval and action safely

10. Multimodal AI capabilities

Multimodal solution landscape

1. Text, image, audio, video, and document understanding
2. Choosing multimodal patterns for business use cases

Using multimodal models in application flows

1. Input handling and normalization
2. Cross-modal reasoning considerations

3. Output design for downstream systems

Complex-content understanding

1. Documents with mixed structure
2. Rich media processing patterns
3. Extracting structured insight from varied inputs

Multimodal pipeline design

1. Ingestion
2. Processing
3. Enrichment
4. Application consumption

11. Complex-content understanding and content extraction

Understanding unstructured and semi-structured content

1. Documents
2. Forms
3. Reports
4. Media-derived artifacts

Content Understanding capabilities

1. Schema-driven extraction
2. Structured output generation
3. Workflow integration

Application design for extracted insights

1. Post-processing and validation
2. Routing to downstream systems
3. Human review patterns where needed

12. Applied app-and-agent engineering practices

Design patterns for real-world delivery

1. Internal productivity assistants
2. Knowledge assistants
3. Action-oriented process agents
4. Human-in-the-loop operational assistants

Performance and scalability considerations

1. Throughput and concurrency

2. Model and tool invocation cost awareness
3. Response-time optimization

Reliability engineering

1. Timeout handling
2. Retry and fallback logic
3. Partial-failure design

Observability and operations

1. Logging and tracing
2. Evaluation in production
3. Change management for prompts, tools, and models

13. Responsible AI and production governance

Responsible AI principles in the delivery lifecycle

1. Fairness, reliability, privacy, transparency, and accountability

Safety controls for generative and agentic systems

1. Input screening
2. Output moderation
3. Access control and human approval gates

Compliance-aware solution design

1. Data handling considerations
2. Enterprise governance checkpoints
3. Documentation and auditability

Production readiness review

1. Technical readiness
2. Operational readiness
3. Risk readiness

14. Capstone architecture synthesis

End-to-end solution framing

1. Defining the application or agent scenario
2. Mapping business goals to Azure components
3. Selecting app, agent, or multi-agent patterns

Delivery blueprint

1. Resource layout

2. Development workflow
3. Evaluation workflow
4. Deployment workflow

Transition to implementation

1. Backlog shaping
2. Team roles
3. Next-step lab or project planning

© 2026 Faysal Aziz · Kuala Lumpur

https://faysalaziz.com/courses/ai-103-accelerator/