

GSDC

Agentic AI Developer Certification

Your 2027 Roadmap to Building Production-Ready AI Agents

From AI fundamentals to deployed, secure, multi-agent systems, a structured path for developers building the next generation of autonomous software.

Why Agentic AI, Why Now

Artificial intelligence is moving past a single-turn, prompt-and-response model. Modern AI systems are increasingly expected to plan multi-step tasks, choose and call tools, retrieve and reason over external information, and carry out workflows with only limited human oversight. That shift — from generation to action — is what "agentic AI" refers to, and it is already reshaping how software gets built.

The pace of adoption is not theoretical. A handful of data points illustrate how quickly this is happening:

- Microsoft reported in 2025 that more than 15 million developers were already using GitHub Copilot, and more than 230,000 organizations had used Copilot Studio to build AI agents and automations.
- JetBrains' 2026 Developer Ecosystem Survey, based on responses from over 15,000 professional developers, found that 90% were using AI coding agents at work at least weekly — with 68% using them daily between May and July 2026.
- The World Economic Forum's Future of Jobs Report 2025 names AI and big data as the fastest-growing skill category through 2030, with AI/ML specialists and software developers both ranking among the fastest-growing job roles.

None of this means every developer must become an AI specialist overnight. It does mean that developers who understand how agents interact with tools, data, and systems — and who can reason about reliability, cost, and safety — are positioned for a rapidly expanding set of roles that did not exist in their current form even two years ago.

Who This Certification Is For

This program is built for people who already think like builders — engineers who want a structured, hands-on path into agentic systems rather than a scattered collection of tutorials and framework docs.

You're a strong fit if you are:

- A software developer who wants to move from writing conventional applications to building systems that plan, decide, and act.
- A machine learning or data engineer who understands models but hasn't yet worked with tool-calling, orchestration, or multi-agent architectures.
- A generative AI or LLM engineer looking to formalize prompt-and-API experience into production-grade agent design.
- A solutions architect or technical lead evaluating how to bring agentic systems into an existing enterprise stack.
- A career switcher with a programming background who wants a credential that demonstrates focused, current AI engineering skill.

Prerequisites

There's no formal prerequisite exam, but you'll get the most out of the program if you're already comfortable with:

- Working proficiency in Python or JavaScript/TypeScript.
- Basic familiarity with REST APIs and how to call them programmatically.

- Exposure to at least one large language model API (OpenAI, Anthropic, or similar) — even simple prompt-response usage is enough of a starting point.

Example: if you've built a chatbot that calls an LLM API and returns a response, but you've never had that chatbot call an external tool, remember a prior conversation, or hand off part of a task to another agent — this certification picks up exactly where that project left off.

Learning Path Overview

The curriculum follows a single, deliberate progression: it starts with how agents think and ends with how they run safely in production. Each module builds directly on the one before it, so the order below is also the order you'll move through the program.

	Agentic AI Foundations Agent types, lifecycles, event-driven design
	Programming Intelligent Agents Goals, planning, reasoning, function calling
	Multi-Agent Systems Delegation, roles, consensus, coordination
	Frameworks, RAG & Tool Integration LangChain, LangGraph, AutoGen, CrewAI
	Memory, State & Context Vector DBs, summarization, persistent state

	Testing, Observability & Deployment
	Debugging, logging, Docker, Kubernetes
	Security & Guardrails
	Prompt-injection defense, human-in-the-loop
	Capstone Projects
	Voice agent + full-stack agentic chatbot

Think of it as a straight line from concept to production: you first learn what an agent is and how it moves through an Understand → Plan → Select tools → Execute → Observe → Evaluate → Adapt cycle, then you learn to build one, then several working together, then how to keep it reliable, and finally how to ship it safely.

Full Curriculum Breakdown

Each module below combines conceptual grounding with hands-on implementation. This is the core of the program — roughly 70% of program time is spent in modules 2 through 7.

Module 1 — Agentic AI Foundations

This module establishes the vocabulary and mental models used throughout the rest of the program. You'll learn the difference between a reactive agent (responds to a single input) and a deliberative agent (plans across multiple steps before acting), and how event-driven programming underlies most modern agent runtimes.

- Agent types: reactive, deliberative, hybrid, and learning agents

- The agent lifecycle: initialization, perception, planning, action, feedback
- Event-driven and reactive programming patterns for agent runtimes
- State management fundamentals and why agents need memory at all
- Orientation to the major frameworks you'll use later: LangChain, AutoGen, CrewAI, Semantic Kernel

Example: you'll trace a simple customer-support agent through its full lifecycle — from receiving a ticket, to deciding whether it needs to look something up, to drafting a reply, to logging the outcome — before writing a single line of framework code.

Module 2 — Programming Intelligent Agents

Here the program moves from concepts to construction. You'll build agents that can set goals, reason about how to reach them, and take action through function calling and tool use.

- Function calling and structured tool/plugin integration
- LLM-based reasoning patterns for goal decomposition
- Vector stores as a foundation for retrieval-driven reasoning
- State graphs for representing multi-step agent workflows
- Planning approaches: ReAct (Reason + Act) and Tree-of-Thought

Example: you'll implement a ReAct-style agent that, given the goal "find and summarize this week's top 3 competitor product launches," decides on its own which searches to run, evaluates the results, and only stops once it has enough information to answer confidently.

Module 3 — Multi-Agent Systems

Real-world workflows are rarely handled by a single agent. This module covers how specialized agents coordinate — dividing labor, communicating results, and resolving disagreements.

- Agent-to-agent communication protocols
- Task delegation and dynamic role assignment
- Synchronization across agents working in parallel
- Conflict resolution when agents produce contradictory outputs
- Consensus mechanisms for multi-agent decision-making

Example: a research-and-writing pipeline where a "researcher" agent gathers sources, a "fact-checker" agent verifies claims, and a "writer" agent drafts the final output — each with a distinct role, and a defined process for what happens when the fact-checker flags something the researcher missed.

Module 4 — Agent Frameworks, RAG & Tool Integration

This module gives you hands-on depth in the frameworks that power most production agentic systems today, along with Retrieval-Augmented Generation (RAG) — one of the most common ways agents ground their answers in real, current information.

- LangChain and LangGraph for building stateful, graph-based agent workflows
- AutoGen and CrewAI for multi-agent orchestration
- LLM integration patterns: APIs, function calling, plugin architecture
- Retrieval-Augmented Generation: chunking, embedding, retrieval, and re-ranking
- Connecting agents to external tools — search, databases, internal company systems

Example: you'll build a RAG-backed support agent that retrieves relevant passages from a product's documentation before answering, rather than relying purely on what the underlying model already "knows" — reducing the chance of a confident but wrong answer.

Module 5 — Memory, State & Context

An agent that forgets everything between turns is severely limited. This module covers how to give agents both short-term working memory and long-term persistent memory, without letting context windows spiral out of control.

- Short-term vs. long-term memory architectures
- Vector databases for semantic memory retrieval
- Context-window management and summarization strategies
- Checkpoints and caching for efficient, resumable workflows
- Knowledge graphs and persistent state stores

Example: an onboarding assistant that remembers a specific user's role, prior questions, and preferences across sessions — spread over days or weeks — without re-sending the entire conversation history on every call.

Module 6 — Testing, Observability & Deployment

Building an agent is the easy part; knowing whether it actually works, and keeping it running once it's live, is where most projects succeed or fail. This module treats agents as production software, not demos.

- Testing and simulation strategies specific to non-deterministic agent behavior
- Debugging techniques and behavioral logging

- Observability: tracing an agent's decisions step by step
- Hallucination analysis and detection
- Deployment with Docker and Kubernetes; cloud deployment patterns
- API design considerations: rate limiting, retries, and reliability mechanisms

Example: you'll instrument an agent with step-by-step tracing so that when it gives a wrong answer, you can replay exactly which tool calls, retrieved documents, and reasoning steps led to that output — rather than guessing.

Module 7 — Security & Guardrails

Because agents can take actions and access tools — not just generate text — security stops being optional. This module covers how to constrain what an agent is allowed to do, and how to catch it when something goes wrong.

- Prompt-injection and jailbreak testing
- Secure tool execution and permission scoping
- Guardrails and output filtering
- Traceability and audit logging for compliance
- Fallback strategies when an agent fails or is uncertain
- Human-in-the-loop controls for high-stakes actions

Example: designing an agent that can draft a refund but requires human sign-off before it actually processes one over a certain dollar amount — a guardrail pattern you'll implement directly, not just discuss.

Module 8 — Hands-On Capstone Projects

The program concludes with two applied projects that combine everything from Modules 1–7 into deployable systems you can show to an employer or client.

- Voice-agent development: a spoken-interface agent that plans, calls tools, and responds in real time
- Full-stack agentic chatbot: a complete, deployed multi-agent application with memory, tool access, and monitoring built in

These are not toy exercises — both projects are built to the same standard covered in Module 6, meaning they're expected to be observable, testable, and deployable, not just functional on your laptop.

Tools & Frameworks Covered

Across the program, you'll get direct, hands-on experience with the tools most in demand for agentic AI roles today:

- LangChain — general-purpose framework for chaining LLM calls and tools
- LangGraph — graph-based orchestration for stateful, multi-step agents
- AutoGen — multi-agent conversation and coordination framework
- CrewAI — role-based multi-agent orchestration
- Semantic Kernel — Microsoft's framework for integrating AI into applications
- Vector databases — for embeddings, semantic search, and RAG
- Docker & Kubernetes — for containerized deployment and scaling

You won't just read about these — every module includes exercises where you implement the concept in at least one of these frameworks, so by the end of the program you have working familiarity with the tools employers are actually asking for in job listings.

Capstone Projects

Project 1: Voice-Agent Development

You'll design and build an agent with a spoken-language interface — one that has to plan, decide which tools to call, and respond within the tighter latency constraints that real-time voice interaction demands. This project is a strong demonstration of your ability to work with agent architectures under practical performance pressure, not just accuracy.

Project 2: Full-Stack Agentic Chatbot

This is the program's flagship deliverable: a complete, deployed multi-agent chatbot application with persistent memory, external tool access, and production-style monitoring built in from the start. It's designed to function as a portfolio centerpiece — the kind of project you can walk an interviewer through screen by screen.

Why this matters: certifications validate that you were taught the material. Projects prove you can build with it. Recruiters and hiring managers increasingly ask for both — a credential to establish baseline knowledge, and a portfolio piece to establish applied skill. These two capstones are built to satisfy both asks at once.

Career Outcomes

Agentic AI skills map to a growing — and still-consolidating — set of job titles. Because "agentic AI developer" is not yet a role tracked separately by government labor statistics, the same skill set shows up under several different titles depending on the employer:

- Agentic AI Developer
- AI Engineer
- Generative AI Engineer
- AI Application Developer
- LLM Engineer
- AI Automation Engineer
- AI Solutions Engineer
- AI Platform Engineer
- AI Agent Architect
- Machine Learning Engineer

Compensation Snapshot (US, 2026 data)

Broad market	\$115K – \$193K / yr	ZipRecruiter & Glassdoor averages, 2026
Enterprise / funded startups	\$150K – \$500K total comp	Engineers building agentic features

Frontier labs / hot AI startups	\$600K – \$1M+ total comp	Small population relative to broader market
---------------------------------	---------------------------	---

A few additional data points worth noting: agentic AI specialization tends to carry a 30–50% salary premium over generalist software engineering roles, and senior multi-agent specialists specifically can command 2–3x a generalist's compensation. Outside the US, India's market ranges from roughly ₹8 LPA for freshers up to ₹1 crore+ for experienced architects, with Germany, Canada, Australia, Singapore, and the UAE also standing out as strong regional markets.

These figures are market snapshots from salary-reporting platforms, not guarantees — actual compensation depends heavily on experience, location, company stage, and the ability to demonstrate production-grade systems, not just certification completion.

Market Growth Context

- Global AI spending is projected to grow 31.9% year-over-year between 2025 and 2029, reaching \$1.3 trillion and over 26% of all worldwide IT spending — driven substantially by agentic AI-enabled applications.
- The agentic AI market specifically is projected to grow from \$7.29B in 2025 to \$139.19B by 2034 — roughly a 19x increase — at a compound annual growth rate of 40.50% between 2026 and 2034.
- Growth is gradual through 2028 (staying under \$20B) before accelerating sharply from 2029 onward, nearly doubling every two years in the back half of the decade.

Certification Logistics

Format

Self-paced online modules combined with live instructor-led sessions for the multi-agent systems and capstone modules — allowing you to move through foundational material on your own schedule while getting direct guidance on the more complex, project-based work.

Duration

Designed to be completed in 8–12 weeks at a part-time pace (6–8 hours/week), or accelerated over 4–6 weeks for full-time learners.

Assessment Structure

- Module-end knowledge checks (auto-graded)
- A practical mid-program assessment covering Modules 1–5
- Final capstone project review and evaluation

Certificate Validity

The certification does not expire, but GSDC recommends staying current through periodic refresher content as frameworks and best practices evolve — this field moves quickly, and the certification is a foundation, not a finish line.

How to Enroll / Next Steps

If you've read this far, you already have a sense of whether this program fits where you want to take your career. Here's how to move forward:

- Visit the GSDC Agentic AI Developer Certification page to review full pricing and start dates.
- Book a short call with an advisor if you want help deciding whether this certification or a different track fits your background.
- Enroll directly if you're ready cohorts open on a rolling basis.

[Enroll Now]

Certification + Agentic AI Skills + Portfolio Projects + Production Experience + Continuous Learning — that combination, not the credential alone, is what will define the strongest agentic AI developers heading into 2027.

AGENTIC AI DEVELOPER CERTIFICATION

LEARN DIRECTLY FROM GLOBAL AI PRACTITIONERS, AGENTIC AI DEVELOPERS, AND INDUSTRY LEADERS WHO ARE SHAPING THE FUTURE OF AUTONOMOUS AI AGENTS AND INTELLIGENT SOFTWARE DEVELOPMENT.



ABOUT GSDC CERTIFICATION



EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.



EBOOK

Extensive and exclusive Ebook created by world's experts to help you with understanding core concepts.



CREATED BY EXPERTS

GSDC certifications are created and authored by world's leading experts in the field.



LEARNING MATERIALS

Get access to learning materials such as videos, ebooks, templates, and practice exams, which will help you clear the certification exam.

LEARNING OBJECTIVE

- Access practical use case studies
- Utilize ready-to-implement expert templates
- Enhance AI project deployment efficiency

Enroll now with the code **LEARN20** To avail **20%** discount

Enroll Now



www.gsdccouncil.org