Certified Agentic AI & Robotics Engineer (CAARE) 2026

Certification Program Syllabus and Study Guide

From AI-Native Software Development to Physical AI & Humanoid Robotics

Colearning Agentic AI with Python — Spec-Driven Reusable Intelligence

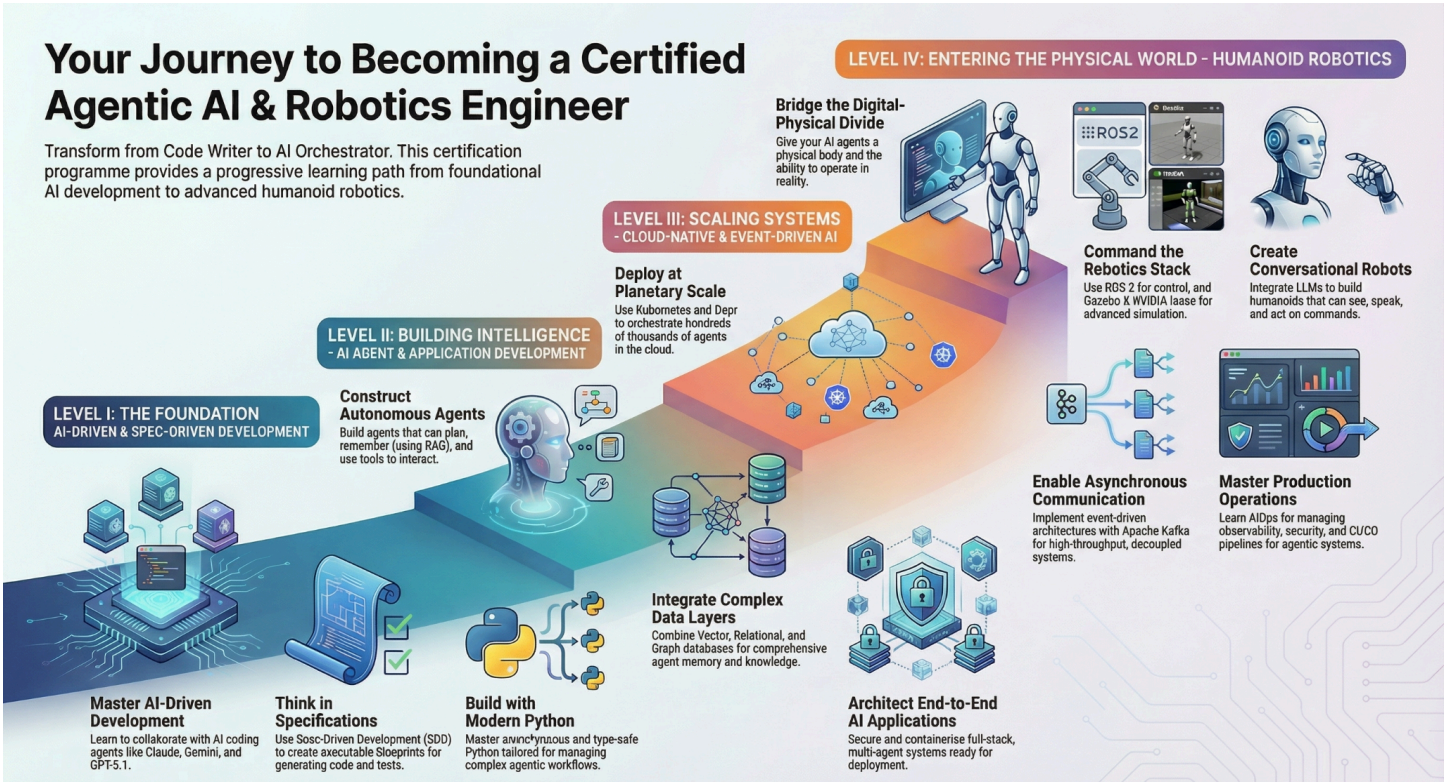
Primary Textbook:

AI Native Software Development Book
https://ai-native.panaversity.org/

Version: 4.2 (December, 2025)

English Video Overview: https://youtu.be/83IXZp4UNcI
Urdu/Hindi Video Overview: https://youtu.be/D2UJSW7LkB4

PIAIC, GIAIC, and Panaversity



Program Overview

The Certified Agentic AI & Robotics Engineer (CAARE) program is a comprehensive journey that transforms learners from beginners to production-ready engineers capable of building intelligent systems across both digital and physical domains. The program covers AI-native software development, multi-agent systems, cloud deployment and AIOps, and culminates with Physical AI and Humanoid Robotics.

The CAARE program is a rigorous certification designed to produce a new breed of engineer capable of bridging the digital and physical worlds. We have moved beyond simple automation into the era of **Agentic Intelligence** and **Physical AI**.

Skill Partnerships: The Future of Work in the Age of AI

The future of work will be a **partnership** between **people**, intelligent **agents** (AI software), and **robots**. This shift won't necessarily eliminate jobs but will change what humans do, leading to a massive demand for new skills.

Key Takeaways:

- **Automation Potential:** Today's technologies could, in theory, automate over **half of the current work hours**. The realization of this potential requires companies to **redesign entire workflows**—not just individual tasks—around this human-AI collaboration.
- **The Skill Shift:** Over 70% of current skills remain relevant but will be applied differently. Humans will spend less time on routine tasks and more time on high-value activities like **framing questions, writing specs, interpreting results, and exercising judgment**.
- **Skill Demand:** The ability to use and manage AI tools, known as **AI fluency**, is the fastest-growing skill requirement for employers.
- **Vulnerable Skills:** Digital and information-processing skills (e.g., accounting, basic coding) are most exposed to automation.
- **Economic Impact:** Successfully implementing these skill partnerships could unlock trillions of dollars in economic value.

As a CAARE candidate, you will:

- **AI-Driven Development:** Learn to collaborate and use AI-Driven Development with coding agents like **Claude Code**, Gemini CLI, GPT-5.1-Codex-Max and the latest Chinese Models.
- **Architect Intelligence**: Master **Spec-Driven Development (SDD)** to co-create complex software systems using **Spec-Kit Plus**.
- **Develop Reusable Intelligence**: Move from reusable code to reusable coding intelligence (horizontal and vertical) using **Coding Subagents** and **Agent Skills**.
- **Build the Brain**: Develop **autonomous agents** using Python, Vector Databases (RAG), Relational, and Graph Databases, and advanced reasoning patterns (**OpenAI/Google/Microsoft/Anthropic Agent SDKs**) using AI-Driven Development.
- **Scale the System**: Deploy planetary-scale infrastructures using **Kubernetes, Dapr, and Kafka** to orchestrate hundreds of thousands of agents using Cloud-Native and AIOps.

- **Embody the Agent**: Give your AI a physical form. You will integrate your agents into **Humanoid Robots using ROS 2 and NVIDIA Isaac™**, creating machines that can see, speak, and act in the human world.

Program Tagline

"From Code Writer to AI Orchestrator."

Master the full spectrum of artificial intelligence: Architecting reasoning software agents in the cloud and deploying embodied humanoid robots in the physical world.

Program Philosophy

The program embraces the co-learning paradigm where humans and AI agents refine each other's understanding. Students don't just use AI as a tool—they learn to collaborate with reasoning entities, write specifications that become executable blueprints, validate AI-generated implementations, develop horizontal and vertical reusable intelligence, and ultimately extend AI capabilities into the physical world through robotics.

The Four Pillars of CAARE

Pillar 1: AI-Driven and Spec-Driven Development

- AI-Driven Development with Claude Code, Gemini CLI, GPT-5.1-Codex-Max and the latest Chinese Models
- Spec-Driven Development methodology using Spec-Kit Plus

Pillar 2: Agentic AI

- Python mastery for AI reasoning and intelligence
- Multi-agent architectures with OpenAI Agents SDK, Google ADK, Microsoft Agent Framework, and Anthropic Agent SDK

Pillar 3: AIOps For Cloud-Native Autonomous Systems

- Cloud-native deployment with Docker, Kubernetes, Dapr, and Ray
- An AI-driven, cloud-native platform built on Docker + Kubernetes + Dapr + Ray, where AI agents control orchestration, scaling, routing, and observability

Pillar 4: Physical AI & Robotics

- ROS 2 (Robot Operating System) for robotic control
- Gazebo and Unity robot simulation
- NVIDIA Isaac™ AI robot development platform
- Humanoid robot design and natural interaction
- Integration of LLMs for conversational robotics

Target Competency Levels

This program prepares students for organizational AI maturity Levels 3-5: Transforming Workflows (AI-Driven and Spec-Driven), Constructing Reusable Coding Intelligence (Coding Subagents and Agent Skills), Building Intelligence (AI-Native Products), and Living in the Future (AI-First Enterprise with Physical AI). Graduates achieve 5-10x productivity in software development and deployment and gain cutting-edge skills in embodied AI systems.

Program Textbook

The CAARE program is built around the comprehensive open source textbook:

AI Native Software Development

Colearning Agentic AI with Python and TypeScript – Spec Driven Reusable Intelligence
https://ai-native.panaversity.org/

This book, alongside the carefully curated curriculum, is designed to build the complete "Full Stack of Intelligence"—from the neural reasoning of an LLM to the kinematic motion of a humanoid robot.

Certification

At the end of each quarter, you will take and pass a certification exam to validate your mastery of the concepts and technologies covered in that quarter. Successful completion of all four levels of certification is required to earn the Certified Agentic AI & Robotics Engineer (CAARE) designation. Students can reattempt the certification exams.

Level I Certification Award: AI Driven and Spec Driven Development

Completion of the Level I certification requires candidates to successfully pass two exams:

1. **Exam L1:P1-AIDD:** AI-Drive and Spec-Driven Development Foundation
2. **Exam L1:P2-PYAI:** Modern Python using AI-Driven Development

Level II Certification Award: Spec-Driven and AI-Native Development

Completion of the Level II certification requires candidates to successfully pass additional one exam:

3. **Exam L2:P1-AIAD:** AI Agent Development
4. **Exam L2:P2-E2EA:** AI-Native End-to-End Application Development

Level III Certification Award: AI Cloud-Native and Event-Driven Development

Completion of the Level III certification requires candidates to successfully pass additional two exams:

5. **Exam L3:P1-CNED:** AI Cloud-Native and Event-Driven Development
6. **Exam L3:P2-AISF;** Agentic AI Startup Founder Self-Study Exam

Level IV Certification Award: Physical AI & Humanoid Robotics

Completion of the Level IV certification requires candidates to successfully pass additional one exams:

7. **Exam L4:P1-PAHR:** Physical AI & Humanoid Robotics

Previous Exam Version Exemptions for New Exams

Those candidates who have passed the previous version of the exams will be exempted from the following new exams.

Previous Exams Version Passed	New Exam Exempted
Level 1 Python	Exam L1:P2-PYAI: Modern Python using AI-Driven Development
Level 2 OpenAI Agents SDK + MCP	Exam L2:P1-AIAD: AI Agent Development

Program Summary

Graduate Competencies

Upon completion, CAARE graduates will be able to:

1. Work as specification engineers and system architects
2. Build AI-native applications with Python
3. Design and implement multi-agent agentic AI systems
4. Deploy enterprise-grade applications to production cloud environments
5. Develop and simulate humanoid robots with ROS 2 and Gazebo
6. Build Physical AI systems using NVIDIA Isaac platform
7. Integrate conversational AI into robotic systems
8. Compete in both digital and physical AI markets as "super orchestrators"

Quarter Progression Map

Qtr	Focus	Key Deliverable
Q1	AI Revolution, Tools & Env	AI-Driven Project
Q2	AI-Driven Python Fundamentals	Python CLI Application
Q3	Spec-Driven AI Agent Dev	Spec-Driven + Agents

Qtr	Focus	Key Deliverable
Q4	AI-First End-to-End App Development	MCP, Better Auth, RAG, FastAPI, OpenAI ChatKit
Q5	AI Cloud-Native and Event Driven	Production Multi-Agent
Q6	Physical AI & Robotics	Humanoid + Conv. AI

The CAARE Advantage

CAARE graduates possess a unique combination of skills that spans the entire AI spectrum—from specification engineering and multi-agent orchestration in digital systems to physical AI and humanoid robotics. This streamlined 5-quarter program delivers maximum impact by focusing on Python-centric AI development and preparing engineers for the next frontier where intelligent systems operate seamlessly across both digital and physical domains.

> *"The future of AI extends beyond the digital space into the physical world."*
>
> — CAARE Program Philosophy

AI-101: Fundamentals of AI-Driven and Spec-Driven Development (Quarter 1)

Book Coverage: Parts 1-4 (Chapters 1-14)

Duration: *13 Weeks*
Focus and Theme: *Paradigm Shift, Strategic Mindset, AI-Native Environment, Tool Mastery, Prompt Engineering, Context Management, and Spec-Driven Methodologies*
Goal: *Forge a "Co-Learning" partnership with AI agents, empowering students to orchestrate intelligent systems. Learners gain mastery over essential tools, development environments, and the core languages of precise prompting, structured context, and reusable specifications for scalable, agentic workflows.*

Quarter Overview

This intensive foundational quarter ignites the paradigm shift from traditional coding to AI-native and spec-driven development, grounding students in the why and how of 2025's $3T developer revolution. Through the nine pillars of AI-Driven Development (AIDD), learners explore economic inflection points, solo-strategy frameworks (e.g., Piggyback Protocol), and the strategic edge of agentic collaboration. Hands-on modules equip setups for AI agents (Claude Code, Gemini CLI), AI-native IDEs (Zed, Cursor), and Git/Bash workflows, while building fluency in Markdown-structured prompts, progressive context loading, and Spec-Driven Development (SDD) for modular, validated intelligence. By quarter's end, students transition from AI consumers to orchestrators, ready to spec and co-build reusable systems with confidence.

- Module 1: The Paradigm Shift
    - Ch 1-4: The AI Revolution, The "Super Orchestrator" Strategy and Mindset, and the 9 Pillars of AI-Driven Development.
- Module 2: The Toolset
    - Ch 5-9: Mastering the Agentic IDEs (Zed, Cursor, Antigravity), Claude Code, and Gemini CLI.
- Module 3: The Language of AI
    - Ch 10: Markdown: Structuring thought for machines.
    - Ch 11: Prompt Engineering: The 8-element framework for precise control.
    - Ch 12: Context Engineering: Managing token limits and memory windows.
- Module 4: SDD-RI Fundamentals
    - Ch 13-14: The Spec-Kit Plus methodology. Writing executable specifications that allow AI to generate code, tests, and docs.

Learning Outcomes

1. Understand the $3 trillion developer economy transformation
1. Master the nine pillars of AI-Driven Development
2. Apply the "Snakes and Ladders" competitive framework
3. Configure and operate AI CLI tools (Claude Code, and Gemini CLI)
4. Understand Coding Subagents and Agent Skills
5. Master AI-First IDEs (Zed, Cursor, Antigravity)
6. Set up and Learn Linux development environment with WSL and DevContainers
7. Learn Bash and Github for AI-Driven Development
8. Apply prompt engineering and context engineering techniques
9. Learn SDD-RI Fundamentals

Weekly Breakdown

Week 1: The AI Development Revolution

- Chapter 1: The AI Development Revolution
- Chapter 2: AI Turning Point — Evidence of the 2025 inflection
- Economic forces, ICPC results, and enterprise reorganization patterns

Week 2: Strategic Opportunities & Nine Pillars

- Chapter 3: How to Make a Billion Dollars in the AI Era
- Chapter 4: The Nine Pillars of AI-Driven Development
- Snakes and Ladders framework, PPP strategy, super orchestrators

Weeks 3-5: AI Tools Setup and Operations

- Chapter 5: Setting up our Main AI Coding Companion (Claude Code) and how to operate in detail. How to build subagents and agent skills.

Weeks 6-8: IDEs & Development Environment

- Chapter 7: Linux Universal Development Environment and Bash for AI-Driven Development
- Chapter 8: AI-First IDEs — Zed, Cursor, and Antigravity
- Chapter 9: Git and Github for AI-Driven Development

Weeks 9-11: AI Communication Mastery

- Chapter 10: Prompt Engineering — Effective communication with AI
- Chapter 11: Context Engineering — Providing optimal context
- The co-learning feedback loop: clarity, reflection, evolution

Weeks 12-13: SDD-RI Fundamentals

- Chapter 13: Understanding Spec-Driven Development with Reusable Intelligence
- Chapter 14: Spec-Kit Plus Hands-On

Assessments

- Two Mid-Term Quizzes in Class
- Strategic Analysis Paper — Apply the AI-Driven frameworks to write a paper on a vertical market (Education, Healthcare, Marketing and Sales, etc.).
- Environment Setup Verification (Pass/Fail)
- Prompt Engineering, Subagent, and Agent Skills Portfolio
- Q1 Midterm Capstone: Complete Personal Website project using Claude Code, Gemini CLI, and Next.js. The project should be published on Vercel.
- Q1 Capstone: Complete Book project using Claude Code, Gemini CLI, and Docusaurus. The book should be published on Github Pages. The book tile can be any one of the following:
  1. Physical AI & Humanoid Robotics
  2. Spec-Drive Development
  3. Reusable Coding Intelligence

Certification Exam

Exam L1:P1-AIDF: AI-Drive and Spec-Driven Development Foundation
Level 1 Certification Exam
Duration: 2 Hours | Questions: 90

AI-151: Modern Python with AI Co-Learning (Quarter 2)

Book Coverage: Parts 5 (Chapters 15-32)

Duration: *13 Weeks*
Focus and Theme: *Python Programming Through AI Collaboration. Learning the Logic of Agents.*
Goal: *A deep dive into modern Python, specifically tailored for asynchronous agentic workflows and type-safe reliability.*

Quarter Overview

Students master Python—the core language of AI-native development—through the AI-Driven Development approach: understand concepts, express intent, let AI execute, validate results. This is not traditional syntax memorization; it's learning to think in specifications while building real applications with AI assistance.

- Module 1: Modern Python Stack
  - Ch 15: Setting up the "Lightning Stack" (uv, Zed, Ruff).
- Module 2: Operators and Data Types
  - Ch 16-19: Data types, operators, key words, and variables.
- Module 3: Control Flow and Loops
  - Ch 20: if, elif, else, for, and while.
- Module 4: Data & Structure
  - Ch 21-26: Advanced Data Structures, Type Hinting, and Functional Patterns suitable for AI data streams.
- Module 5: Object-Oriented Programming
  - Ch 27-29: Classes, Objects, Metaclasses and Dataclasses.
- Module 6: Pydantic and Generics
  - Ch 30: Pydantic (runtime validation that catches bad data) and Generics (static type safety that catches errors before runtime).
- Module 7: Asynchronous Mastery
  - Ch 31: AsyncIO & Concurrency: The critical skill for managing multiple parallel LLM streams.
- Module 8: Free-Threaded Python
  - Ch 32: Python 3.14 made free-threading production-ready. For the first time in Python's history, you can now write truly parallel multi-threaded Python code that scales across CPU cores. A 4-agent AI system running on a 4-core machine can achieve 2–4x performance gains instead of pseudo-concurrency.

Learning Outcomes

1. Master Python project management with UV package manager
2. Understand data types, operators, and control flow
3. Write functions with type hints and documentation
4. Work with data structures: lists, dictionaries, sets, tuples
5. Implement object-oriented programming patterns
6. Handle errors and implement test-driven development
7. Develop multi-threaded Python programs

Weekly Breakdown

Week 1: Python Project Setup

- Chapter 15: Python UV — The Fastest Python Package Manager
- Project structure, dependency management, virtual environments
- Professional Python development workflow

Weeks 2-3: Data Types & Variables

- Chapters 16 - 17: Data Types — Integers, Floats, Strings, Booleans
- Chapter 18: Operators, Keywords, and Variables
- Chapter 19: Strings and Type Casting
- Type hints as communication with AI

Weeks 4-5: Operators & Control Flow

- Chapter 20: Comparison Operators — True/False decisions
- Logical Operators — and, or, not
- Control Flow — if/elif/else, loops, pattern matching

Weeks 6-7: Functions & Data Structures

- Chapter 21-22: Data Structures — Lists, tuples, dictionaries, sets
- Chapter 23: Functions — Definition, parameters, return values
- List comprehensions and functional programming

Week 8: Error Handling

- Chapter 24: Error Handling

Weeks 9-11: OOP & Testing

- Chapter 27-29: Object-Oriented Programming — Classes, inheritance
- Test-Driven Development with pytest

Weeks 12-13: AsyncIO & Free-Threaded Python

- Chapter 31: AsyncIO & Concurrency
- Chapter 32: Free-Threaded Python

Assessments

- Weekly coding challenges with AI collaboration
- Data structures project
- OOP design challenge
- Q2 Capstone: Full Python CLI applications with comprehensive tests:
    - Todo CLI App
    - Calculator CLI App
    - Basic Notes CLI App
    - Weather CLI App (API Integration)

Certification Exam

Exam L1:P2-PYAI: Modern Python using AI-Driven Development
Level 1 Certification Exam
Duration: 2 Hours | Questions: 90

Level I Certification Award

Candidates will receive their Level I certification after successfully passing two required exams:

1. **Exam L1:P1-AIDF:** AI-Drive and Spec-Driven Development Foundation
2. **Exam L1:P2-PYAI:** Modern Python using AI-Driven Development

AI-321: AI Agent Development with Spec-Driven Development (Quarter 3)

Book Coverage: Parts 6

Duration: *13 Weeks*
Focus and Theme: *Agent Development and Deep Intelligence.*
Goal: *Building the first functional autonomous agents.* Moving beyond simple chatbots to agents that can plan, remember (RAG), and use tools to interact with the world.

Quarter Overview

This pivotal quarter introduces the core methodology that transforms developers into specification engineers. Students master Spec-Kit Plus methodology, write clear testable specifications, and build their first AI agents. This is where the paradigm shift becomes practical.

- Module 1: The Major Frameworks
  - Agent Fundamentals (Loops, Perception, Action).
  - OpenAI Agents SDK.
- Module 2: Tool Use (MCP)
  - Model Context Protocol (MCP): Connecting agents to external data and APIs.
- Module 3: Agent Quality & Architecture
  - Test-Driven Development (TDD) and Evals for Agents.
  - Advanced Patterns: Reflection, Planning, and Multi-Agent Orchestration.
- Module 4: The Data Layer (Memory)
  - Vector Databases: Implementing Retrieval Augmented Generation (RAG).
  - Relational and Graph Databases for structured agent knowledge.

Learning Outcomes

1. Understand specifications as living contracts
2. Write specifications
3. Implement Spec-Kit Plus methodology with Reusable Intelligence
4. Design and build AI agents
5. Understand agent architecture patterns and tool integration

Weekly Breakdown

Weeks 1-5: Building Agents with OpenAI SDK

- OpenAI Agents SDK Deep Dive
- Building Your Agents
- Tool integration, function calling, and state management

Weeks 6-7: Building MCP Servers with FastMCP

- Understand MCP fundamentals
- Develop custom MCP servers
- Implement code execution capabilities for more efficient agents

Weeks 8-9: Building Effective Agents with Design Patterns

- Advanced Agent Architectures: Exploring planning, reflection, and multi-agent patterns.
- Combo Pattern: Composing multiple agents and tools into sophisticated workflows.
- Applying established software design patterns to agent construction for maintainability, scalability, and reusability.

Week 10-11: Data Layer

- Add persistence and knowledge retrieval.
- Integrate vector databases for RAG, relational databases for structured data, and graph databases for relationship-rich knowledge

Week 12: Building Conversational UI with OpenAI ChatKit

- Integrating OpenAI ChatKit for seamless, interactive agent conversations.
- Managing conversational flow, displaying agent thought processes, and user feedback loops.
- Creating rich, dynamic user experiences for agent-driven applications.OpenAI Agents SDK Deep Dive

Week 13: Quality Practices

- Validate agent correctness and reliability.
- Apply test-driven development patterns and implement comprehensive evaluation frameworks to ensure agents behave as specified.

Assessments

- Spec-Driven Todo 3-tier app
- SDD Single-agent Chatbot with OpenAI Agents SDK, RAG, and ChatKit
- Q3 Capstone: SDD Multi-agent system

Certification Exam

Exam L2:P1-AIAD: AI Agent Development
Level 2 Certification Exam
Duration: 2 Hours | Questions: 90

AI-351: Build AI-First End-to-End Projects using SDD, Better Auth, RAG, FastAPI, MCP, ChatKit, DataStores, and Cloud Containers (Quarter 4)

Book Coverage: Integration Focus

Duration: 13 Weeks
Focus and Theme: Full-Stack Agentic AI Application Development. Integrating Multi-Agent Systems, Security, and Cloud-Native Data Stores.
Goal: Transition from building individual agents (Quarter 3) to architecting, securing, and deploying complete, production-ready, AI-native applications using Spec-Driven Development (SDD) as the single source of truth.

Quarter Overview

This intensive project-based quarter serves as the critical bridge between **Agent Development (AI-321)** and **Cloud-Native Deployment (AI-451)**. Students will move beyond conceptual agents and implement a full-stack, secure, multi-agent application. The focus is on integrating all previously learned components: using **SDD** to define the entire application, securing it with **Better Auth** principles, providing memory with **RAG** (Vector DBs) and structured data (**PostgreSQL/Neo4j**), exposing services via **FastAPI**, and making agents tool-ready via the **Model Context Protocol (MCP)**. Developing ChatBots/Front-end with OpenAI ChatKit. This quarter prepares the student to become a production-ready **"Super Orchestrator."**

- Module 1: **SDD for End-to-End Systems**
    - Designing the Master Specification for an entire multi-service, multi-agent application.
    - Specifying APIs, database schemas, agent responsibilities, and deployment requirements.
- Module 2: **Better Auth and Security for Agents**
    - Implementing modern authentication (e.g., OAuth 2.0, JWT) for agent-as-a-service APIs.
    - Securing tool calls (MCP) and data layer access.
- Module 3: **Deep Data Integration**
    - Mastering the integration of **Vector Databases (RAG)** for unstructured knowledge.
    - Implementing **Relational (PostgreSQL)** and **Graph (Neo4j)** databases for structured and relationship-rich agent memory.
- Module 4: **Chatbot/Front-End**
    - Build chatbot UI with OpenAI ChatKit
- Module 5: **Project Finalization and Readiness**
    - Wrapping agent logic into secure, performant microservices using **FastAPI**.
    - Creating the **Docker** image
    - Deploy Docker image to Hugging Face

Learning Outcomes

1. Architect an entire AI-Native application from a single, master **Spec-Driven Development (SDD)** blueprint.
2. Implement robust **Authentication and Authorization** for agent-driven microservices.
3. Design and integrate a mixed data-layer combining **Vector, Relational, and Graph Databases** for comprehensive agent memory.

4. Develop **Model Context Protocol (MCP)** servers to enable agents to use external APIs and data stores as secure, high-value tools.
5. Containerize a full multi-agent system and prepare it for **Cloud-Native (Kubernetes)** deployment.

Weekly Breakdown

Week(s)	Focus	Key Topics
1-2	**Project Specification & Arch**	Writing the Master SDD Spec. Decomposing the system into microservices and defining agent APIs (FastAPI/Pydantic models).
3-4	**API-First Development & Auth**	Implementing **FastAPI** agents. Integrating OAuth 2.0/JWT for secure API endpoints (**Better Auth**).
5-7	**Deep Data: RAG & Relational**	Setting up a **PostgreSQL** (or similar) database for structured data. Integrating a **Vector DB** for Retrieval Augmented Generation (RAG).
8-9	**Graph Data Integration**	Introduction to **Graph Databases (Neo4j)**. Modeling complex relationships and integrating a graph-based knowledge store into a dedicated agent.
10-11	**MCP Tool & Agent Interop**	Building a dedicated **FastMCP** server to expose the data layer and other agents as secure tools to the main coordinating agent.

12	**Chatbot UI**	Use **OpenAI ChatKit**
13	**Containerization & Deployment Prep**	Writing production-ready **Dockerfiles** for all services. Creating basic **Kubernetes** manifests and validating the system locally with Docker Compose.

Assessments

- **Mid-Term Quiz:** Architecture and Security of Multi-Agent Systems.
- **SDD Master Spec Portfolio:** A complete, validated specification for the Capstone Project.
- **Q4 Capstone:** A fully functional, authenticated, and containerized **"AI-Native Project Manager"** application that:
  - Uses a **Vector DB** for document search.
  - Uses a **Relational DB** for project/task status.
  - Uses a coordinating agent to manage user requests via a secure **FastAPI** endpoint.

Certification Exam

Exam L2:P2-E2EA: AI-Native End-to-End Application Development
Level 2 Certification Exam
Duration: 2 Hours | Questions: 90

Level II Certification Award

Candidates will receive their Level II certification after successfully passing two required exams:

1. **Exam L2:P1-AIDD:** AI Agent Development
2. **Exam L2:P2-E2EA:** AI-Native End-to-End Application Development

AI-451: AI Cloud-Native and Event-Driven Development (Quarter 5)

Book Coverage: Part 7

Duration: 13 Weeks
Focus and Theme: Multi-Agent Systems & Production Deployment. From Scripts to Services.
Goal: Containerizing agents and enabling asynchronous communication for

high-throughput systems. Planetary Scale: Deploying massive clusters of agents that can maintain state and handle enterprise loads.

Quarter Overview

Students advance to building sophisticated multi-agent systems and deploying them to production. This quarter covers multi-agent coordination, retrieval-augmented generation, and the complete cloud-native deployment stack: Docker, Kubernetes, Dapr, and Ray. By the end, students can deploy enterprise-grade AI systems.

- Module 1: Service Wrappers
  - FastAPI: Exposing agents as RESTful microservices.
- Module 2: Containerization
  - Docker: Building optimized, portable containers for AI applications.
- Module 3: Orchestration
  - Kubernetes: Deploying, scaling, and managing agent clusters.
- Module 4: The Dapr Framework
  - Dapr (Distributed Application Runtime):
  - Actors: The perfect model for digital twins and stateful agents.
  - Pub/Sub: Managing signals between services.
  - Workflows: Orchestrating long-running business processes.
- Module 5: Event-Driven Architecture
  - Apache Kafka: Decoupling agent communication. Moving from linear request/response to event streams (essential for robotics where sensor data flows continuously).
- Module 6: Operations
  - Observability, Security, and API Gateways.
- Module 7: Infrastructure
  - CI/CD pipelines and Infrastructure-as-Code.

Learning Outcomes

1. Master API-First architecture principles
2. Build agents with Google Agent Development Kit (ADK)
3. Containerize applications with Docker
4. Deploy and manage with Kubernetes
5. Scale AI Agentic workloads with Dapr
6. Scale AI workloads with Ray

Weekly Breakdown

Weeks 1-2: API-First & Specifications

- API-First Architecture
- Developing APIs with FastAPI and SSD

Weeks 2-3: Docker & Containerization

- Chapter 41: Docker Fundamentals
- Dockerfile best practices and multi-stage builds

- Container networking and security

Weeks 4-9: Kubernetes Deployment

- Kubernetes Architecture
- Deploying AI Applications to Kubernetes
- Pods, Services, Deployments, ConfigMaps

Weeks 10-13: Distributed Systems & Production

- Dapr — Distributed Application Runtime
- Ray for Distributed AI
- Production Operations and Monitoring

Assessments

- Multi-agent FastAPI system implementation
- Docker containerization project
- Kubernetes deployment challenge
- Q4 Capstone: Production-deployed multi-agent AI system

Certification Exams

Exam L3:P1-CNED: AI Cloud-Native and Event-Driven Development
Level 3 Certification Exam
Duration: 2 Hours | Questions: 90

Exam L3:P2-AISF Agentic AI Startup Founder Self-Study Exam
Level 3 Certification Exam
Duration: 4 Hours | Questions: 162

Exam L3:P2-AISF Self-Study Materials:

You should emphasise innovation in your studies because our main goal is innovation. When you study these papers look from the perspective of how it will help you innovate.

The Piggyback Protocol Pivot (PPP) Strategy: A Strategic Framework for Market Entry and Disruption in Established Industries Through Agentic AI
https://docs.google.com/document/d/1aFuB7VMgD4JjUVj0h-pVPubZowA2weNLan-w58TXIsk/edit?usp=sharing

The Complete Guide to Building Agentic AI Startups: Lean, Design Thinking, and Agile
https://docs.google.com/document/d/1Zu90L8WWe76h7FKJkHYwg6XhkUJG2TldiTsKvVzbrjw/edit?usp=sharing

The Comprehensive Guide to Funding Agentic AI Startups: From Bootstrapping to Venture Capital
https://docs.google.com/document/d/1IbfFp0XWlz71HOnK0zQBPgwShEy8zvVd_l04D3N3lws/edit?usp=sharing

Comprehensive Business Tutorial for Agentic AI Startup Founders
https://docs.google.com/document/d/1jZ8kcjoBUDmShFuurkGRGjnvv8Tkv9P4LkzOxnlypX8/edit?usp=sharing

Agentic AI Startup Founder: The Comprehensive Crash Course
https://docs.google.com/document/d/15SZH6TiPD2yY3zcqx7lhGV7DlnAY6gl1TYAZgdrPupY/edit?usp=sharing

Level III Certification Award

Candidates will receive their Level III certification after successfully passing two required exams:

1. **Exam L3:P1-CNED:** AI Cloud-Native and Event-Driven Development
2. **Exam L3:P2-AISF:** Agentic AI Startup Founder Self-Study Exam

AI-551: Physical AI & Humanoid Robotics (Quarter 6)

Extended Curriculum: Embodied AI & Robotics

Book Coverage: Part 13

Duration: 13 Weeks
Focus and Theme: AI Systems in the Physical World. Embodied Intelligence.
Goal: Bridging the gap between the digital brain and the physical body. Students apply their AI knowledge to control Humanoid Robots in simulated and real-world environments.

Quarter Overview

The future of AI extends beyond digital spaces into the physical world. This capstone quarter introduces Physical AI—AI systems that function in reality and comprehend physical laws. Students learn to design, simulate, and deploy humanoid robots capable of natural human interactions using ROS 2, Gazebo, and NVIDIA Isaac.

- Module 1: The Robotic Nervous System (ROS 2)
  - Focus: Middleware for robot control.
  - ROS 2 Nodes, Topics, and Services.
  - Bridging Python Agents (Q2/Q3) to ROS controllers using rclpy.
  - Understanding URDF (Unified Robot Description Format) for humanoids.

- Module 2: The Digital Twin (Gazebo & Unity)
  - Focus: Physics simulation and environment building.
  - Simulating physics, gravity, and collisions in Gazebo.
  - High-fidelity rendering and human-robot interaction in Unity.
  - Simulating sensors: LiDAR, Depth Cameras, and IMUs.

- Module 3: The AI-Robot Brain (NVIDIA Isaac™)
  - Focus: Advanced perception and training.
  - NVIDIA Isaac Sim: Photorealistic simulation and synthetic data generation.
  - Isaac ROS: Hardware-accelerated VSLAM (Visual SLAM) and navigation.
  - Nav2: Path planning for bipedal humanoid movement.

- Module 4: Vision-Language-Action (VLA)
  - Focus: The convergence of LLMs and Robotics.
  - Voice-to-Action: Using OpenAI Whisper for voice commands.
  - Cognitive Planning: Using LLMs to translate natural language ("Clean the room") into a sequence of ROS 2 actions.
  - Capstone Project: The Autonomous Humanoid. A final project where a simulated robot receives a voice command, plans a path, navigates obstacles, identifies an object using computer vision, and manipulates it.

Why Physical AI Matters

Humanoid robots are poised to excel in our human-centered world because they share our physical form and can be trained with abundant data from interacting in human environments. This represents a significant transition from AI models confined to digital environments to embodied intelligence that operates in physical space.

Learning Outcomes

1. Understand Physical AI principles and embodied intelligence
2. Master ROS 2 (Robot Operating System) for robotic control
3. Simulate robots with Gazebo and Unity
4. Develop with NVIDIA Isaac AI robot platform
5. Design humanoid robots for natural interactions
6. Integrate GPT models for conversational robotics

Weekly Breakdown

Weeks 1-2: Introduction to Physical AI

- Foundations of Physical AI and embodied intelligence
- From digital AI to robots that understand physical laws
- Overview of humanoid robotics landscape
- Sensor systems: LIDAR, cameras, IMUs, force/torque sensors

Weeks 3-5: ROS 2 Fundamentals

- ROS 2 architecture and core concepts
- Nodes, topics, services, and actions
- Building ROS 2 packages with Python
- Launch files and parameter management

Weeks 6-7: Robot Simulation with Gazebo

- Gazebo simulation environment setup
- URDF and SDF robot description formats
- Physics simulation and sensor simulation
- Introduction to Unity for robot visualization

Weeks 8-10: NVIDIA Isaac Platform

- NVIDIA Isaac SDK and Isaac Sim
- AI-powered perception and manipulation
- Reinforcement learning for robot control
- Sim-to-real transfer techniques

Weeks 11-12: Humanoid Robot Development

- Humanoid robot kinematics and dynamics
- Bipedal locomotion and balance control
- Manipulation and grasping with humanoid hands
- Natural human-robot interaction design

Week 13: Conversational Robotics & Final Capstone

- Integrating GPT models for conversational AI in robots

- Speech recognition and natural language understanding
- Multi-modal interaction: speech, gesture, vision
- Final CAARE Program Capstone Presentation

Assessments

- ROS 2 package development project
- Gazebo simulation implementation
- Isaac-based perception pipeline
- Q5 Capstone: Simulated humanoid robot with conversational AI
- Final CAARE Capstone: End-to-end demonstration of agentic + physical AI

Hardware Requirements

This course is technically demanding. It sits at the intersection of three heavy computational loads: **Physics Simulation** (Isaac Sim/Gazebo), **Visual Perception** (SLAM/Computer Vision), and **Generative AI** (LLMs/VLA).

Because the capstone involves a "Simulated Humanoid," the primary investment must be in **High-Performance Workstations**. However, to fulfill the "Physical AI" promise, you also need **Edge Computing Kits** (brains without bodies) or specific robot hardware.

1. The "Digital Twin" Workstation (Required per Student)

This is the most critical component. NVIDIA Isaac Sim is an Omniverse application that requires "RTX" (Ray Tracing) capabilities. Standard laptops (MacBooks or non-RTX Windows machines) **will not work**.

- **GPU (The Bottleneck):** NVIDIA **RTX 4070 Ti (12GB VRAM)** or higher.
  - *Why:* You need high VRAM to load the USD (Universal Scene Description) assets for the robot and environment, plus run the VLA (Vision-Language-Action) models simultaneously.
  - *Ideal:* RTX 3090 or 4090 (24GB VRAM) allows for smoother "Sim-to-Real" training.
- **CPU:** Intel Core i7 (13th Gen+) or AMD Ryzen 9.
  - *Why:* Physics calculations (Rigid Body Dynamics) in Gazebo/Isaac are CPU-intensive.
- **RAM: 64 GB DDR5** (32 GB is the absolute minimum, but will crash during complex scene rendering).
- **OS: Ubuntu 22.04 LTS**.
  - *Note:* While Isaac Sim runs on Windows, ROS 2 (Humble/Iron) is native to Linux. Dual-booting or dedicated Linux machines are mandatory for a friction-free experience.

2. The "Physical AI" Edge Kit

Since a full humanoid robot is expensive, students learn "Physical AI" by setting up the *nervous system* on a desk before deploying it to a robot. This kit covers Module 3 (Isaac ROS) and Module 4 (VLA).

- **The Brain: NVIDIA Jetson Orin Nano** (8GB) or **Orin NX** (16GB).
    - *Role:* This is the industry standard for embodied AI. Students will deploy their ROS 2 nodes here to understand resource constraints vs. their powerful workstations.
- **The Eyes (Vision): Intel RealSense D435i** or **D455**.
    - *Role:* Provides RGB (Color) and Depth (Distance) data. Essential for the VSLAM and Perception modules.
- **The Inner Ear (Balance):** Generic USB IMU (BNO055) (Often built into the RealSense D435i or Jetson boards, but a separate module helps teach IMU calibration).
- **Voice Interface:** A simple USB Microphone/Speaker array (e.g., ReSpeaker) for the "Voice-to-Action" Whisper integration.

3. The Robot Lab

For the "Physical" part of the course, you have three tiers of options depending on budget.

Option A: The "Proxy" Approach (Recommended for Budget)

Use a quadruped (dog) or a robotic arm as a proxy. The software principles (ROS 2, VSLAM, Isaac Sim) transfer 90% effectively to humanoids.

- **Robot: Unitree Go2 Edu** (~$1,800 - $3,000).
- **Pros:** Highly durable, excellent ROS 2 support, affordable enough to have multiple units.
- **Cons:** Not a biped (humanoid).

Option B: The "Miniature Humanoid" Approach

Small, table-top humanoids.

- **Robot: Unitree H1** is too expensive ($90k+), so look at **Unitree G1** (~$16k) or **Robotis OP3** (older, but stable, ~$12k).
- **Budget Alternative: Hiwonder TonyPi Pro** (~$600).
    - *Warning:* The cheap kits (Hiwonder) usually run on Raspberry Pi, which **cannot** run NVIDIA Isaac ROS efficiently. You would use these only for kinematics (walking) and use the Jetson kits for AI.

Option C: The "Premium" Lab (Sim-to-Real specific)

If the goal is to actually deploy the Capstone to a real humanoid:

- **Robot: Unitree G1 Humanoid**.
    - *Why:* It is one of the few commercially available humanoids that can actually walk dynamically and has an SDK open enough for students to inject their own ROS 2 controllers.

4. Summary of Architecture

To teach this successfully, your lab infrastructure should look like this:

Component	Hardware	Function
Sim Rig	PC with RTX 4080 + Ubuntu 22.04	Runs Isaac Sim, Gazebo, Unity, and trains LLM/VLA models.
Edge Brain	Jetson Orin Nano	Runs the "Inference" stack. Students deploy their code here.
Sensors	RealSense Camera + Lidar	Connected to the Jetson to feed real-world data to the AI.
Actuator	Unitree Go2 or G1 (Shared)	Receives motor commands from the Jetson.

If you do not have access to RTX-enabled workstations, we must restructure the course to rely entirely on cloud-based instances (like AWS RoboMaker or NVIDIA's cloud delivery for Omniverse), though this introduces significant latency and cost complexity.

Building a "Physical AI" lab is a significant investment. You will have to choose between building a physical **On-Premise Lab at Home** (High CapEx) versus running a **Cloud-Native Lab** (High OpEx).

Option 2 High OpEx: The "Ether" Lab (Cloud-Native)

Best for: Rapid deployment, or students with weak laptops.

1. Cloud Workstations (AWS/Azure) Instead of buying PCs, you rent instances.

- **Instance Type:** AWS **g5.2xlarge** (A10G GPU, 24GB VRAM) or **g6e.xlarge**.
- **Software:** NVIDIA Isaac Sim on Omniverse Cloud (requires specific AMI).
- **Cost Calculation:**
    - Instance cost: ~$1.50/hour (spot/on-demand mix).
    - Usage: 10 hours/week × 12 weeks = 120 hours.
    - Storage (EBS volumes for saving environments): ~$25/quarter.
    - **Total Cloud Bill: ~$205 per quarter**.

2. Local "Bridge" Hardware You cannot eliminate hardware entirely for "Physical AI." You still need the edge devices to deploy the code physically.

- **Edge AI Kits:** You still need the Jetson Kit for the physical deployment phase.

    - **Cost: $700** (One-time purchase).
- **Robot:** You still need one physical robot for the final demo.
    - **Cost: $3,000** (Unitree Go2 Standard).

The Economy Jetson Student Kit

Best for: Learning ROS 2, Basic Computer Vision, and Sim-to-Real control.

Component	**Model**	**Price (Approx.)**	**Notes**
The Brain	**NVIDIA Jetson Orin Nano Super Dev Kit (8GB)**	**$249**	New official MSRP (Price dropped from ~$499). Capable of 40 TOPS.
The Eyes	**Intel RealSense D435i**	**$349**	Includes IMU (essential for SLAM). Do not buy the D435 (non-i).
The Ears	**ReSpeaker USB Mic Array v2.0**	**$69**	Far-field microphone for voice commands (Module 4).
Wi-Fi	(Included in Dev Kit)	$0	The new "Super" kit includes the Wi-Fi module pre-installed.
Power/Misc	SD Card (128GB) + Jumper Wires	$30	High-endurance microSD card required for the OS.
TOTAL		**~$700 per kit**	

3. The Latency Trap (Hidden Cost)

- Simulating in the cloud works well, but *controlling* a real robot from a cloud instance is dangerous due to latency.
- *Solution:* Students train in the Cloud, download the model (weights), and flash it to the local Jetson kit.

Certification Exam

Exam L4:P1-PAHR: Physical AI & Humanoid Robotics
Level IV Certification Exam
Duration: 2 Hours | Questions: 90

Certified Agentic AI & Robotics Engineer Award

Candidates will receive their **CAARE** certification after successfully passing one required exams:

1. **Exam L4:P1-PAHR:** Physical AI & Humanoid Robotics