UDACITY
Part of **Accenture**

SCHOOL OF ARTIFICIAL INTELLIGENCE

Agentic AI

Syllabus

UDACITY.COM

Overview

Go beyond single chatbots to engineer sophisticated, coordinated teams of AI agents. This Nanodegree guides you from advanced prompting techniques like Chain-of-Thought and ReAct to designing agentic workflows with patterns like Routing and Parallelization. You'll master building and orchestrating agents in Python that can reason, plan, and use tools to interact with databases and external APIs. Build a powerful portfolio by tackling hands-on projects, including a multi-agent travel planner, an AI-powered project manager, and a fully automated sales system, to solve real-world problems.

Nanodegree Program | Intermediate | 51 hours | ★ 4.8 (40 Reviews)

Prerequisites

Prior to enrolling, you should have the following knowledge:

OpenAI API | API fluency | Basic Prompting | Basic Python | Generative AI Fluency

You will also need to be able to communicate fluently and professionally **in written and spoken English**.

Skills You'll Learn

Prompting for Effective LLM Reasoning and Planning

Iterative Prompt Development | Implementing Role-Based Prompting | Feedback Loop Design | Persona-Based Prompt Design | Implementing Prompt Chains | Systematic Prompt Refinement | CoT and ReAct | Agentic Reasoning Frameworks | Prompt Chain Design | Implementing Feedback Loops

Agentic Workflows

Agentic Task Routing Design | Implementing Agentic Workflows | Implementing Agent Orchestration | Agentic Iterative Refinement Workflow Design | Agentic Workflow Design & Visualization | Implementing Parallel Agent Workflows | Implementing Sequential Task Prompt Chain Workflows | Agent Orchestration Design | Sequential Task Decomposition for Prompt Chaining | Implementing Reflective AI Workflows | Agentic Parallel Workflow Design

Building Agents

Implementing API Integration for Agents | Implementing Single-agent RAG | Implementing AI Agent Evaluation | State Management for AI Agents | API Integration for Agents | Implementing Long-term Memory Management for AI Agents | Agent Tool Use | Database Interaction for AI Agents | Long-term Memory Management for AI Agents | Implementing State Management for AI Agents | Agent Memory Design | Implementing Agent Tool Use | Web Search Integration & Data Retrieval for AI Agents | Agent Memory Design with Python and OpenAI | Evaluating AI Agents | Implementing Database Interaction for AI Agents | Implementing LLM Structured Outputs | Single-Agent Retrieval Augmented Generation | Implementing Web Search Integration and Data Retrieval for AI Agents | LLM Structured Outputs

Multi-Agent Systems

Multi-Agent System Design | Multi-Agent Implementation | Multi-Agent State Management Concepts | Implementing Multi-Agent Orchestration | Multi-Agent Systems | Implementing Multi-Agent State Coordination | Multi-Agent Agent Orchestration Concepts | Implementing Multi-Agent Routing | Multi-Agent State Coordination Concepts | Implementing Multi-Agent RAG | Multi-Agent Routing Concepts | Implementing Multi-Agent State Management | Multi-Agent RAG Concepts

Courses

01. Prompting for Effective LLM Reasoning and Planning

13 hours

Go beyond basic chatbots and learn to engineer sophisticated AI agents. Learn advanced prompting techniques that power modern AI. You'll master Chain-of-Thought, ReAct, and feedback loops to build systems that can reason, plan, and solve complex problems. Through hands-on exercises, you will transform generic AI into specialized, reliable tools, culminating in building a multi-agent travel planner from scratch.

L1	Introduction to Prompting for Effective LLM Reasoning and Planning	Introduces the core concepts of Agentic AI, the course structure, prerequisites, and learning environment.
L2	The Role of Prompting in Agentic AI with Python and OpenAI	Learn what AI Agents are and how they work. Understand the critical role prompting plays in guiding them to reason, plan, and act to achieve goals.
L3	Role-Based Prompting	Explains the theory of using roles or personas to control the tone, style, and expertise of an LLM's output.
L4	Implementing Role-Based Prompting with Python	Provides hands-on practice in iteratively developing a role-based prompt to create a believable historical figure persona.
L5	Chain-of-Thought and ReACT Prompting	Explains the conceptual frameworks for Chain-of-Thought (CoT) for guided reasoning and ReAct (Reason+Act) for enabling agents to plan and take actions.
L6	Applying COT and ReACT Prompting with Python	Provides hands-on practice implementing both CoT and ReAct prompts to solve a retail analytics problem.
L7	Prompt Instruction Refinement	Explains the theory of systematically refining prompt instructions by modifying components like Role, Task, Context, Examples, and Output Format.
L8	Applying Prompt Instruction Refinement with Python	Provides hands-on practice iteratively refining a prompt to transform a generic recipe analyzer into a precise dietary consultant that produces structured JSON.
L9	Chaining Prompts for Agentic Reasoning	Explains the conceptual framework for building multi-step AI workflows by linking the output of one prompt to the input of the next, and the importance of validation.
L10	Chaining Prompts with Python	Provides hands-on practice implementing a three-stage prompt chain with Pydantic-based gate checks to automate an insurance claim triage process.

L11	LLM Feedback Loops	Explains the conceptual framework for building self-improving systems where an agent uses feedback from its own actions to iteratively refine its output.
L12	Implementing LLM Feedback Loops with Python	Provides hands-on practice building an automated feedback loop where an AI generates Python code, has it tested against a unit test suite, and uses the test results as feedback to debug itself.
L13	Congratulations!	Course review
L14	Project: Project: AgentsVille Trip Planner: A Multi-Agent Travel Assistant System	In this project, you'll build an agentic travel assistant system, the "AgentsVille Trip Planner"

02. Agentic Workflows

13 hours

Go beyond simple automation and learn to architect intelligent systems. In this course, you'll master the art of designing and building agentic workflows using Python. You'll explore core patterns like Prompt Chaining, Routing, and Parallelization to create teams of AI agents that can reason, plan, and act to solve complex problems. You will finish by building a complete, agentic project management system, proving your ability to translate high-level goals into powerful, adaptive AI solutions.

L1	Introduction to Agentic Workflows	Introduces the foundational concepts of AI agents and agentic workflows, setting the stage for the course. It covers prerequisites, the course environment, and how to use the necessary API keys.
L2	Understanding Agentic Workflows	Explores what defines a modern AI agent, its core components (Persona, Knowledge, Tools, Interaction), and the different types of agents based on their LLM interaction model.
L3	Agentic Workflow Modeling	Design and visualize agentic workflows. Learn common agent types as building blocks for creating visual workflow diagrams.
L4	Agentic Workflow Implementation	Covers the practical aspects of translating agentic workflow models into Python code. Students learn to structure agent logic, define agent classes, and orchestrate their interactions.
L5	Agentic Workflow Patterns: Prompt Chaining Workflow	Introduces the Prompt Chaining pattern for breaking down complex tasks into a sequence of smaller, dependent steps. It covers strategies for task decomposition, validation, and context management.
L6	Implementing Agentic Prompt Chaining Workflows with Python	Provides hands-on experience in implementing the Prompt Chaining pattern. Students build a multi-agent chain to solve a problem where information is passed sequentially.

L7	Agentic Workflow Patterns: Routing	Teaches the Routing pattern, which involves classifying incoming tasks and directing them to the most appropriate specialized agent or processing path.
L8	Implementing Agentic Routing Workflows with Python	Students implement a routing system where a router agent uses an LLM to classify a query and then dispatches it to the correct specialist agent, which may involve orchestrating sub-tasks.
L9	Agentic Workflow Patterns: Parallelization	Introduces the Parallelization pattern for executing multiple agent tasks concurrently. It covers strategies for task decomposition (sharding, aspect-based) and result aggregation.
L10	Implementing Agentic Parallelization Workflows with Python	Students implement a parallel workflow using Python's threading module, where multiple specialist agents analyze a document concurrently, and a synthesizer agent combines their findings.
L11	Agentic Workflow Patterns:- Evaluator-Optimizer Workflow	Focuses on the Evaluator-Optimizer pattern, an iterative process of generation, critique, and refinement to improve output quality. It emphasizes clear evaluation criteria and actionable feedback.
L12	Implementing Agentic Evaluator-Optimizer Workflows with Python	Students build a two-agent system (a creator and a critic) that works in a loop. The creator generates a solution, and the critic provides feedback until the solution meets all constraints.
L13	Agentic Workflow Patterns Orchestrator-Workers Workflow	Introduces the advanced Orchestrator-Workers pattern, where a central agent dynamically plans, delegates, and synthesizes the work of multiple specialized worker agents.
L14	Implementing Agentic Orchestrator-Workers Pattern in Python	Students implement a market analysis report generator where an Orchestrator agent creates a plan, assigns tasks to news, competitor, and trend analysis workers, and then synthesizes their findings.
L15	Course Review	Course review.
L16	Project: AI-Powered Agentic Workflow for Project Management	In this project you'll build a comprehensive, reusable library of different agent types and then use them to create a multi-step agentic workflow to manage a technical project.

03. Building Agents

11 hours

Build robust AI agents. Integrate tools via function calling, generate structured outputs with Pydantic, manage agent state, and utilize short-term and long-term memory. Create data-driven agents that interact with external APIs, search the web, query SQL databases, and perform agentic RAG for dynamic retrieval. Learn to evaluate agent performance for reliable, real-world applications.

L1	Introduction to Building Agents	Get to know your course instructors, set up OpenAI resources, and get an overview of the course.
L2	Extending Agents with Tools	Extend AI agents beyond text with tool integrations, enabling reliable real-time actions and data access.
L3	Building Agents with Tools in Python	Develop AI agents in Python using tools with OpenAI SDK. Interact through language models, build functionality-enhancing tools, and test via tool-augmented exercises.
L4	Structured Outputs	Discover structured outputs in AI: transform responses into actionable JSON for integration. Utilize schemas, parsers, and function calls to enhance reliability and automation in workflows.
L5	Implementing Structured Outputs with Pydantic	Master structured outputs with Pydantic and OpenAI SDK for LLMs. Learn parsing, type validation, and create validated AI agent responses in JSON format.
L6	Agent State Management	Explore agent state management with state machines. Learn how agents track user input, instructions, and tool use for complex workflows, ensuring adaptability and reliability.
L7	Implementing Agent State with Python	Master Python state machines: set up environment, define schemas, manage transitions, and run workflows. Explore advanced routing and loops for dynamic workflows.
L8	Short-Term Agent Memory	Explore short-term memory in AI agents, enhancing coherence via state, ephemeral, and ephemeral memory strategies for efficient context retention in active sessions.
L9	Adding Agent Memory with Python	Learn to implement short-term memory in Python for coherent AI interactions via a ChatBot with personas, enabling session continuity and dynamic responses.
L10	External Tools and APIs	Explore using external APIs for real-time data, dynamic actions, and authenticating agents. Discover MCP, a protocol standardizing AI's tool interoperability and safety.
L11	Integrating External Tools and APIs with OpenAI & Python	Explore using OpenAI and Python to integrate external APIs, make GET/POST/PUT requests, manage API keys, and create agents for real-time data interactions.
L12	Web Search Agents	Equip agents to search web for real-time, unstructured info. Ground responses in evidence using APIs, handle noise, and avoid hallucination for credible answers.
L13	Creating Web Search Agents with Python	Build a web search agent using Python, Tavily API, to integrate real-time web data, parse results, and enhance language models' effectiveness.

L14	Interacting with Databases	Equip agents to access and modify structured data by using SQL for interaction and vector databases for semantic tasks, ensuring seamless integration with private systems.
L15	Building Database Agents in Python	Convert natural language to SQL using SQLAlchemy, SQLite, and text2SQL Agent to interact with databases efficiently through real-world examples and practical applications.
L16	Agentic Retrieval Augmented Generation	Discover Agentic RAG: Enhance RAG by enabling reflection, query reformulation, and intelligent adaption for nuanced answers. Master retrieval, reasoning, and retry loops.
L17	Agentic RAG with Python and ChromaDB	Explore agentic RAG in Python using ChromaDB, integrating AI with retrieval-augmented generation for intelligent document retrieval and processing with OpenAI embeddings.
L18	Long-Term Agent Memory	Explore long-term agent memory: understand semantic, episodic, and procedural memories. Learn storage strategies and best practices for personalized, coherent interactions.
L19	Maintaining Long-Term Agent Memory in Python	Implement long-term memory in Python agents using vector databases for enhanced user interaction, session persistence, and personalized responses.
L20	Agent Evaluation	Agent Evaluation guides assessing an agent’s task completion, quality, tool use, and system metrics using response, step, or trajectory strategies to ensure reliable and efficient operations.
L21	Evaluating Agents with Python	Evaluate Python-based agents by setting environments, creating tools, designing test cases, and using diverse evaluation methods to enhance performance and design.
L22	Course Conclusion	Congratulations on completing the course!
L23	Project: UdaPlay - An AI Research Agent for the Video Game Industry	In this project, students will build a stateful AI Research Agent designed to explore the video game industry.

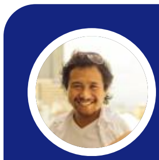
04. Multi-Agent Systems

13 hours

Move beyond single chatbots and learn to build coordinated teams of AI agents. This course takes you through the entire process of creating multi-agent systems, from architectural design to implementation in Python. You will master how to orchestrate complex workflows, manage data flow and state, and use advanced techniques like multi-agent RAG. Through hands-on exercises and a final project building an automated sales system, you will gain the skills to develop powerful, practical agentic AI solutions.

L1	Introduction to Multi-Agent Systems	Learn the core concepts of multi-agent systems and their real-world parallels.
L2	Designing Multi-Agent Architecture	Explain the core components of multi-agent systems and how to design their high-level architecture.
L3	Implementing Multi-Agent Architecture with Python	Develop a multi-agent system by coding the designed architecture and connecting agents with well-defined interfaces.
L4	Orchestrating Agent Activities	Apply orchestration techniques to coordinate multiple agent actions and achieve complex workflows.
L5	Implementing Agent Orchestration	Apply orchestration patterns (sequential parallel conditional) to build a multi-agent system that handles complex workflows with multiple steps and decision points.
L6	Routing and Data Flow in Agentic Systems	Configure routing mechanisms to manage data flow among agents in multi-agent systems.
L7	Implementing Routing and Data Flow in Agentic Systems	Implement a routing agent that can intelligently direct user requests to specialized agents based on the content and urgency of the request.
L8	State Management in Multi-Agent Systems	Evaluate methods for tracking and updating agent state across multi-turn interactions.
L9	Implementing State Management in Multi-Agent Systems	Evaluate methods for tracking and updating agent state across multi-turn interactions.
L10	Multi-Agent Orchestration and State Coordination	Develop a coordinated multi-agent system that synchronizes states for coherent task execution.
L11	Implementing Multi-Agent Orchestration and State Coordination	Implement a multi-agent system that manages concurrent access to shared resources using state coordination techniques to detect and resolve conflicts.
L12	Multi-Agent Retrieval Augmented Generation	Extend RAG to multiple cooperating agents, each specialized in certain retrieval tasks.
L13	Implementing Multi-Agent Retrieval Augmented Generation	Build a multi-agent RAG system with specialized retrieval agents and a synthesis agent to combine information from multiple sources and make a complex judgment.
L14	Course Review	Course review
L15	Project: The Beaver's Choice Paper Company Sales Team	Design and build a complete multi-agent system for a real-world business scenario incorporating architecture orchestration state management and routing.

Meet Your Instructors



Brian Cruz

Head of Core AI

Brian Cruz is the Head of Data and AI Engineering at Advocate, where generative AI is used to help disabled individuals access the government benefits they deserve. Previously he led Machine Learning Engineering for Einstein Guidance at Salesforce, and he also established the Center of Excellence for AI and ML at Samba TV. He studied Mathematics at UC Berkeley.

Peter Kowalchuk

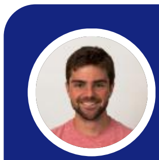
Engagement Director at C3.ai

Peter Kowalchuk is an Engagement Director at C3.ai and an Adjunct Professor at CUNY, where he teaches data science, AI, and information systems. He holds a B.S. in Electrical Engineering, an MBA, and a Master's in Data Science. With 20+ years' experience, he worked at Microsoft on Industry Cloud AI and led AI efforts while working in the energy sector.

Henrique Santana

Principal Machine Learning Engineer at Dell Technologies

Henrique Santana is a Principal Machine Learning Engineer at Dell Technologies, specializing in AI implementation for digital businesses. With a strong background in Problem Solving with Data, he has extensive experience in data engineering, AI operations, and the development of autonomous AI agents. He is also a professor in MBA programs, teaching AI for business leaders. Passionate about technology, he explores Generative AI, LLM frameworks, and emerging AI architectures.



Joshua Bernhard

Staff Data Scientist at Marketplace

Josh has been sharing his passion for data for over a decade. He's used data science for work ranging from cancer research to process automation. He recently has found a passion for solving data science problems within marketplace companies.

Christopher Agostino

Founder and Research Scientist at NPC Worldwide

Chris researches novel AI and NLP technologies using Bayesian methods at NPC Worldwide and builds agents for production at Celeria. He earned his PhD in astronomy from Indiana University and pioneered AI tools in HubSpot's People team.

Why Udacity

Demonstrate proficiency with practical projects

Projects are based on real-world scenarios and challenges, allowing you to apply the skills you learn to practical situations, while giving you real hands-on experience

- ✓ Gain proven experience
- ✓ Retain knowledge longer
- ✓ Apply new skills immediately

24/7 access to real human support

Reviewers provide timely and constructive feedback on your project submissions, highlighting areas of improvement and offering practical tips to enhance your work

- ✓ Get help from subject matter experts
- ✓ Gain valuable insights and improve your skills
- ✓ Learn industry best practices