Advanced Python with AI: Agentic AI, LLMs, and Generative AI Course Curriculum

Course Overview

This advanced-level course is designed for students with a solid foundation in Python programming who wish to delve into the cutting-edge fields of Artificial Intelligence, specifically focusing on Agentic AI, Large Language Models (LLMs), and Generative AI. The curriculum aims to provide both theoretical understanding and practical skills necessary to design, develop, and deploy intelligent AI systems. Upon completion, students will be equipped to tackle complex AI challenges, innovate with new AI technologies, and pursue international certifications in these advanced domains.

Target Audience

- Experienced Python developers and programmers.
- Data Scientists and Machine Learning Engineers looking to specialize in Agentic AI, LLMs, and Generative AI.
- Researchers and academics interested in the practical applications and architectural considerations of advanced AI systems.
- Professionals seeking to upskill and remain competitive in the rapidly evolving AI landscape.

Course Goal

To empower advanced students with the knowledge and practical expertise to build, deploy, and manage sophisticated AI systems leveraging Agentic AI principles, Large Language Models, and Generative AI techniques, preparing them for real-world applications and relevant international certifications.

Course Structure

The course will be divided into several modules, each focusing on a specific area of Advanced AI. The progression is designed to build foundational knowledge before moving to more complex and integrated topics.

Module 1: Foundations of Advanced AI & Python for AI

Learning Objectives:

- Review and master advanced Python programming concepts relevant to AI development, including decorators, context managers, metaclasses, and asynchronous programming.
- Understand the current landscape of Artificial Intelligence, differentiating between traditional AI, Machine Learning, Deep Learning, and the emerging fields of Generative AI and Agentic AI.
- Familiarize with essential Python libraries for AI (e.g., NumPy, Pandas, Scikit-learn, TensorFlow/PyTorch) and best practices for their efficient use.
- Set up a robust development environment for advanced AI projects, including virtual environments, Docker, and basic cloud integration concepts.

Key Topics:

- **Advanced Python for AI:**
  - Decorators for function modification and aspect-oriented programming.
  - Context Managers (`with` statement) for resource management.
  - Generators and Iterators for efficient data processing.
  - Metaclasses and advanced object-oriented programming patterns.
  - Asynchronous Python (`asyncio`) for concurrent AI operations.
  - Performance optimization techniques in Python (profiling, Cython, Numba).

- **AI Landscape Overview:**
  - Evolution of AI: From symbolic AI to connectionism.
  - Machine Learning vs. Deep Learning: Key differences and applications.
  - Introduction to Generative AI: What it is and its impact.
  - Introduction to Agentic AI: Concepts, motivations, and potential.
- **Essential AI Libraries:**
  - NumPy for numerical operations and array manipulation.
  - Pandas for data manipulation and analysis.
  - Brief overview of Scikit-learn for traditional ML algorithms.
  - Introduction to TensorFlow and PyTorch for deep learning (installation and basic tensor operations).
- **Development Environment Setup:**
  - Virtual environments (venv, conda) for dependency management.
  - Introduction to Docker for reproducible environments.
  - Basic Git for version control and collaborative development.

Resources:

- Official Python Documentation for advanced features.
- Selected chapters from "Fluent Python" by Luciano Ramalho.
- Online tutorials and documentation for NumPy, Pandas, TensorFlow, PyTorch.
- Docker documentation and introductory guides.

Module 2: Large Language Models (LLMs) - Deep Dive

Learning Objectives:

- Understand the fundamental architecture of Large Language Models, including Transformers and attention mechanisms.
- Explore various types of LLMs (e.g., GPT, BERT, T5) and their respective strengths and applications.
- Master prompt engineering techniques for effective interaction with LLMs, including few-shot, zero-shot, and chain-of-thought prompting.
- Learn about fine-tuning LLMs for specific tasks and domains, including data preparation and evaluation metrics.
- Understand the ethical considerations and biases inherent in LLMs and strategies for mitigation.

Key Topics:

- **Transformer Architecture:**
  - Encoder-Decoder structure.
  - Self-attention mechanism and multi-head attention.
  - Positional encoding.
- **LLM Types and Applications:**
  - Generative LLMs (e.g., GPT series) for text generation, summarization, translation.
  - Discriminative LLMs (e.g., BERT) for text classification, sentiment analysis.
  - Hybrid models.
- **Prompt Engineering:**
  - Basic prompting: instructions, examples.
  - Advanced prompting techniques: role-playing, persona, constraints.
  - Few-shot, zero-shot, and one-shot learning.
  - Chain-of-thought, tree-of-thought, and other reasoning prompts.

    - Prompt optimization and evaluation.
- **LLM Fine-tuning:**
    - Transfer learning concepts.
    - Data collection and annotation for fine-tuning.
    - Parameter-Efficient Fine-Tuning (PEFT) methods (e.g., LoRA).
    - Evaluation metrics for fine-tuned models.
- **Ethical Considerations and Bias:**
    - Bias in training data and model outputs.
    - Fairness, accountability, and transparency in LLMs.
    - Responsible AI development practices.

Resources:

- "Attention Is All You Need" paper (Transformer architecture).
- Hugging Face Transformers library documentation and tutorials.
- OpenAI API documentation and best practices for prompting.
- Research papers on LLM fine-tuning and ethical AI.

Module 3: Generative AI - Principles and Applications

Learning Objectives:

- Understand the core principles behind various Generative AI models, including GANs, VAEs, and Diffusion Models.
- Explore the applications of Generative AI beyond text, such as image generation, video synthesis, and music composition.
- Learn to implement and experiment with different generative models using Python libraries and frameworks.

- Address the challenges and ethical implications of creating and deploying generative models, including deepfakes and intellectual property.

Key Topics:

- **Generative Model Architectures:**
  - **Generative Adversarial Networks (GANs):** Generator and Discriminator, training process, common architectures (DCGAN, CycleGAN).
  - **Variational Autoencoders (VAEs):** Encoder-Decoder structure, latent space, sampling.
  - **Diffusion Models:** Denoising diffusion probabilistic models (DDPMs), stable diffusion, image generation process.
- **Applications of Generative AI:**
  - **Image Generation:** From text, image-to-image translation, style transfer.
  - **Video Synthesis:** Frame generation, motion transfer.
  - **Music and Audio Generation:** MIDI generation, raw audio synthesis.
  - **Code Generation:** AI-assisted coding, natural language to code.
  - **Data Augmentation:** Generating synthetic data for training other models.
- **Implementation with Python:**
  - Using libraries like `diffusers`, `torchvision`, `Keras-GAN`.
  - Training generative models on custom datasets.
- **Challenges and Ethics in Generative AI:**
  - Model collapse, mode collapse in GANs.
  - Computational resources and training time.
  - Misinformation and deepfakes.

    - Copyright and intellectual property issues.
    - Bias and fairness in generated content.

Resources:

- Original papers on GANs, VAEs, and Diffusion Models.
- Hugging Face `diffusers` library documentation.
- PyTorch and TensorFlow official tutorials on generative models.
- Articles and discussions on the ethical implications of generative AI.

Module 4: Introduction to Agentic AI & Core Concepts

Learning Objectives:

- Define Agentic AI and understand its distinction from traditional AI and machine learning.
- Grasp the core components of an AI agent: perception, reasoning, action, and memory.
- Explore different types of agent architectures, including reactive, deliberative, and hybrid agents.
- Understand the concept of agent loops and their role in autonomous behavior.
- Familiarize with the principles of goal-oriented behavior and planning in AI agents.

Key Topics:

- **What is Agentic AI?**
    - Definition and characteristics of AI agents.
    - Comparison with traditional AI, ML, and LLMs.
    - The shift from models to agents.
- **Core Components of an AI Agent:**
    - **Perception:** How agents gather information from their environment.

    - **Reasoning:** Decision-making processes, knowledge representation.
    - **Action:** How agents interact with their environment.
    - **Memory:** Short-term and long-term memory, knowledge bases.
- **Agent Architectures:**
    - **Reactive Agents:** Simple stimulus-response behavior.
    - **Deliberative Agents:** Planning, goal-directed behavior, symbolic AI concepts.
    - **Hybrid Agents:** Combining reactive and deliberative approaches.
- **Agent Loops:**
    - Perceive-Think-Act cycle.
    - Iterative refinement and self-correction.
- **Goal-Oriented Behavior and Planning:**
    - Defining goals and sub-goals.
    - Planning algorithms (e.g., A*, STRIPS).
    - Task decomposition and execution.

Resources:

- "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig (relevant chapters on agents).
- Online articles and research papers on Agentic AI fundamentals.
- Conceptual diagrams illustrating agent architectures and loops.

Module 5: Building Agentic Systems with Python & Frameworks

Learning Objectives:

- Implement basic AI agents from scratch in Python to understand their internal workings.

- Utilize the OpenAI Agents SDK for building sophisticated AI agents, including tool integration and function calling.
- Integrate LLMs as the reasoning core of AI agents.
- Develop custom tools and functions for agents to interact with external APIs and systems.
- Understand and implement memory management strategies for agents, including short-term and long-term memory.

Key Topics:

- **Agent Implementation from Scratch:**
  - Building a simple agent loop.
  - Implementing basic perception and action mechanisms.
  - State management for agents.
- **OpenAI Agents SDK:**
  - Introduction to the SDK and its components.
  - Defining agent capabilities and tools.
  - Function calling and tool use with LLMs.
  - Handling agent responses and errors.
- **LLMs as Agent Brains:**
  - Connecting LLMs for reasoning and decision-making.
  - Prompt design for agentic behavior.
  - Managing context windows and token usage.
- **Tool Integration and External Systems:**
  - Creating custom Python tools for agents.

    - Interacting with REST APIs, databases, and other services.
    - Handling structured and unstructured data for tool inputs/outputs.
- **Memory Management for Agents:**
    - Short-term memory (context window, scratchpad).
    - Long-term memory (vector databases, knowledge graphs).
    - Retrieval Augmented Generation (RAG) for knowledge retrieval.

Resources:

- learn-agentic-ai GitHub Repository (specifically modules on OpenAI Agents SDK).
- OpenAI Agents SDK documentation.
- LangChain or LlamaIndex documentation for RAG and memory concepts.
- Examples of API integrations (e.g., weather API, search API).

Module 6: Multi-Agent Systems & Collaboration

Learning Objectives:

- Understand the principles and benefits of multi-agent systems (MAS).
- Explore different communication protocols and coordination mechanisms for agents.
- Design and implement multi-agent architectures for collaborative problem-solving.
- Address challenges in MAS such as conflict resolution, trust, and emergent behavior.
- Familiarize with Agent2Agent (A2A) communication and its role in complex agentic ecosystems.

Key Topics:

- **Introduction to Multi-Agent Systems:**
    - Definition and characteristics of MAS.

    - Advantages of using multiple agents (parallelism, robustness, specialization).
    - Use cases for MAS (e.g., supply chain, smart grids, simulations).
- **Agent Communication:**
    - Communication languages (e.g., FIPA ACL).
    - Message passing and shared memory approaches.
    - Model Context Protocol (MCP) for standardized tool use and data exchange.
- **Coordination and Collaboration:**
    - Centralized vs. decentralized coordination.
    - Negotiation and auction protocols.
    - Teamwork and joint intentions.
    - Emergent behavior in MAS.
- **Agent2Agent (A2A) Communication:**
    - Understanding the need for direct agent-to-agent interaction.
    - Implementing A2A patterns for seamless information flow.
    - Security and authentication in A2A.
- **Challenges in MAS:**
    - Conflict detection and resolution.
    - Trust and reputation mechanisms.
    - Scalability of communication and coordination.
    - Debugging and monitoring multi-agent interactions.

Resources:

- Research papers on multi-agent systems and coordination.

- Dapr documentation (for concepts related to distributed systems and messaging).
- Case studies of real-world multi-agent system implementations.
- learn-agentic-ai GitHub Repository (specifically modules on AI protocols and DACA).

Module 7: Scalability, Deployment, and Ethical AI Agents

Learning Objectives:

- Understand the challenges and strategies for scaling AI agents and multi-agent systems to handle large numbers of concurrent users or tasks.
- Learn to deploy AI agents using cloud-native technologies such as Docker, Kubernetes, and Dapr.
- Explore best practices for monitoring, logging, and managing AI agent deployments in production environments.
- Deepen understanding of ethical considerations in AI, specifically for autonomous agents, including safety, fairness, transparency, and accountability.
- Develop strategies for building trustworthy and safe agent architectures.

Key Topics:

- **Scalability Challenges in Agentic AI:**
  - Handling high concurrency and throughput.
  - Resource management for LLM inference and agent operations.
  - Distributed state management and data consistency.
- **Cloud-Native Deployment:**
  - **Docker:** Containerization of AI agents and their dependencies.
  - **Kubernetes:** Orchestration of containerized agents, scaling, self-healing, and resource allocation.
  - **Dapr (Distributed Application Runtime):**

    - Introduction to Dapr and its building blocks (service invocation, state management, pub/sub, actors).
    - Using Dapr for resilient and scalable agent communication and state persistence.
    - Dapr Agents and Workflows for orchestrating complex agent behaviors.
- **Monitoring and Management:**
    - Logging and tracing for agent interactions and decisions.
    - Performance monitoring and bottleneck identification.
    - CI/CD pipelines for AI agent deployment.
- **Ethical AI Agents:**
    - **Safety:** Preventing unintended harmful actions.
    - **Fairness:** Mitigating bias and ensuring equitable outcomes.
    - **Transparency and Explainability:** Understanding agent decisions and actions.
    - **Accountability:** Assigning responsibility for agent behavior.
    - **Trustworthy AI Architectures:** Staged execution, reversible actions, safety patterns.
    - Regulatory landscape and compliance for AI systems.

Resources:

- Docker and Kubernetes official documentation.
- Dapr documentation and examples.
- learn-agentic-ai GitHub Repository (specifically modules on DACA deployment guide and scalability discussions).
- NIST AI Risk Management Framework, IEEE Ethically Aligned Design.

Module 8: Advanced Topics & Future Trends in AI

Learning Objectives:

- Explore cutting-edge research areas and emerging trends in Agentic AI, LLMs, and Generative AI.
- Understand the concepts of embodied AI and its implications for agent development.
- Discuss the potential impact of quantum computing and neuromorphic computing on future AI systems.
- Analyze the societal and economic implications of advanced AI, including job displacement and new opportunities.
- Engage in critical thinking about the future trajectory of AI and its role in human society.

Key Topics:

- **Emerging Trends in Agentic AI:**
  - Self-improving agents and meta-learning.
  - Human-Agent teaming and collaboration.
  - Agent simulation and synthetic environments.
  - Cognitive architectures for agents.
- **Advanced LLM Concepts:**
  - Long-context LLMs and infinite context windows.
  - Multimodal LLMs (integrating vision, audio, etc.).
  - Small Language Models (SLMs) and their efficiency.
  - Knowledge distillation and model compression.
- **Frontiers in Generative AI:**
  - Generative models for 3D content and virtual worlds.
  - Controllable generation and fine-grained control.
  - Generative AI for scientific discovery (e.g., drug design, material science).

- **Embodied AI:**
  - Robotics and physical agents.
  - Interaction with the real world.
  - Sensory perception and motor control.
- **Impact of Future Computing Paradigms:**
  - Quantum AI: Potential for exponential speedups.
  - Neuromorphic computing: Brain-inspired architectures.
- **Societal and Economic Implications:**
  - Future of work and automation.
  - Ethical governance of advanced AI.
  - AI for social good.
  - Global competition and collaboration in AI.

Resources:

- Recent research papers from top AI conferences (NeurIPS, ICML, ICLR, AAAI).
- Reports from leading AI research institutions (e.g., OpenAI, DeepMind, Google AI).
- Books and articles on the philosophy and future of AI.

Module 9: Project Work & Capstone

Learning Objectives:

- Apply all acquired knowledge and skills to design, develop, and implement a comprehensive AI project.
- Work independently or in teams to solve a real-world problem using Agentic AI, LLMs, and/or Generative AI.

- Demonstrate proficiency in advanced Python programming, AI model integration, and deployment strategies.
- Present project outcomes, including technical details, challenges faced, and future improvements.
- Develop a portfolio-ready project that showcases their expertise in advanced AI.

Key Activities:

- **Project Proposal:** Students will define a problem, propose an AI-driven solution, and outline the technologies and methodologies to be used.
- **Design and Development:** Iterative development of the AI system, including agent design, LLM integration, generative component creation, and system architecture.
- **Implementation:** Coding the solution in Python, utilizing relevant libraries and frameworks learned throughout the course.
- **Testing and Evaluation:** Rigorous testing of the AI system, performance evaluation, and debugging.
- **Deployment (Optional):** Deploying the project to a cloud environment (e.g., using Docker and Kubernetes) for demonstration.
- **Documentation:** Comprehensive documentation of the project, including design choices, technical specifications, and user guides.
- **Presentation:** A final presentation showcasing the project, its functionalities, and the insights gained.

Project Ideas (Examples):

- **Intelligent Personal Assistant:** An agent that can understand complex queries, interact with multiple APIs, and generate personalized responses or actions.
- **Automated Content Creator:** A system that generates various forms of content (text, images, code) based on high-level prompts, potentially using multi-agent collaboration for different content types.

- **Agent-Based Simulation:** A simulation environment where multiple AI agents interact to model complex systems (e.g., economic models, traffic flow, social dynamics).
- **AI-Powered Research Assistant:** An agent that can conduct deep research, summarize findings, and generate reports on specific topics using LLMs and knowledge graphs.
- **Adaptive Learning Agent:** An agent that personalizes learning paths and generates educational content based on a student's progress and needs.

Assessment:

- Project Proposal (10%)
- Project Implementation and Code Quality (40%)
- Project Documentation (20%)
- Final Presentation (20%)
- Innovation and Complexity (10%)

Resources:

- All course materials from previous modules.
- Access to cloud computing resources (e.g., AWS, Google Cloud, Azure credits).
- Mentorship and guidance from instructors.
- Peer review and collaboration opportunities.

Assessment Framework

The course assessment will be continuous and comprehensive, designed to evaluate both theoretical understanding and practical application of the concepts learned. The assessment components are weighted to reflect their importance in mastering advanced AI skills.

Components:

- **Quizzes and Assignments (30%):** Regular quizzes after each module to test comprehension of theoretical concepts. Assignments will involve coding exercises and small-scale implementations of AI components.
- **Mid-Term Project (20%):** A focused project after Module 4, requiring students to integrate LLM and Generative AI concepts into a practical application.
- **Final Capstone Project (40%):** The culminating project (Module 9) will be the primary assessment, evaluating the student's ability to design, implement, and present a complex AI solution using Agentic AI, LLMs, and Generative AI principles.
- **Participation and Engagement (10%):** Active participation in discussions, peer reviews, and collaborative exercises will be encouraged and graded.

Grading Scale:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- F: Below 60%

International Certification Pathway

Upon successful completion of this course, students will possess the foundational knowledge and practical skills to pursue various international certifications in Agentic AI, LLMs, and Generative AI. While this course does not directly provide these certifications, it serves as a strong preparation. Students are encouraged to research and select certifications that best align with their career goals.

Here are two highly recommended certifications that align well with the content of this course:

- **IBM RAG and Agentic AI Professional Certificate (Coursera):** This certificate program directly focuses on Retrieval Augmented Generation (RAG) and agentic AI, which are

core components of this course. It validates practical skills in building and deploying AI agents for real-world applications.

- **NVIDIA Generative AI with LLMs Certification:** This certification validates foundational concepts and practical skills in developing and integrating Generative AI models, particularly Large Language Models. Given the significant emphasis on LLMs and Generative AI in this course, this certification is an excellent complement.

Guidance for Certification Pursuit:

1. **Review Prerequisites:** Ensure you meet the specific prerequisites for each certification, as some may require prior experience or specific knowledge areas.
2. **Study Official Materials:** Utilize official study guides, practice exams, and documentation provided by the certification bodies.
3. **Hands-on Practice:** Practical experience gained through the course projects will be invaluable for passing performance-based exams.
4. **Community Engagement:** Join online forums and study groups to share knowledge and prepare with peers.
5. **Stay Updated:** The field of AI is rapidly evolving. Continuously update your knowledge and skills to remain current with the latest advancements.

Revised Course Schedule: 8-9 Months (3 days/week, 2 hours/class)

This revised schedule distributes the course content over approximately 8 to 9 months, with classes held 3 days a week for 2 hours each. This totals approximately 6 hours of instruction per week, allowing for in-depth coverage and practical application.

Total Estimated Instruction Time: 204 hours (based on 34 weeks)

Module Breakdown and Estimated Weeks:

Module No.	Module Title	Estimated Weeks	Estimated Hours (Weeks * 6 hours/week)

1	Foundations of Advanced AI & Python for AI	3	18
2	Large Language Models (LLMs) - Deep Dive	5	30
3	Generative AI - Principles and Applications	4	24
4	Introduction to Agentic AI & Core Concepts	4	24
5	Building Agentic Systems with Python & Frameworks	6	36
6	Multi-Agent Systems & Collaboration	5	30
7	Scalability, Deployment, and Ethical AI Agents	4	24
8	Advanced Topics & Future Trends in AI	3	18
9	Project Work & Capstone	4	24
Total		**38**	**228**

Note on Schedule Flexibility:

The total estimated weeks (38) slightly exceeds the 8-9 month target (32-36 weeks). This provides flexibility for deeper dives into specific topics, additional practical exercises, guest lectures, or review sessions. Instructors can adjust the pace based on student progress and interest. The capstone project (Module 9) can also extend beyond the dedicated 4 weeks, allowing students ample time for development and refinement outside of structured class time.

Weekly Breakdown Example (Illustrative):

- **Week 1-3:** Module 1: Foundations of Advanced AI & Python for AI
- **Week 4-8:** Module 2: Large Language Models (LLMs) - Deep Dive

- **Week 9-12:** Module 3: Generative AI - Principles and Applications
- **Week 13-16:** Module 4: Introduction to Agentic AI & Core Concepts
- **Week 17-22:** Module 5: Building Agentic Systems with Python & Frameworks
- **Week 23-27:** Module 6: Multi-Agent Systems & Collaboration
- **Week 28-31:** Module 7: Scalability, Deployment, and Ethical AI Agents
- **Week 32-34:** Module 8: Advanced Topics & Future Trends in AI
- **Week 35-38:** Module 9: Project Work & Capstone (can extend as needed)

This revised schedule ensures comprehensive coverage of all topics while fitting within the specified time constraints, with a slight buffer for enhanced learning experiences. The international certification pathway remains integrated, as the course content directly prepares students for the listed certifications.