Generative AI with Lang graph

Module 1: Introduction to LangGraph and Generative AI

- Overview of Generative AI and LLMs
- Role of LangChain and the evolution to LangGraph
- LangGraph vs traditional LangChain chaining methods
- Understanding stateful and multi-step reasoning in LLM applications
- When to use LangGraph over standard chains or agents

Module 2: LangGraph Core Concepts

- Nodes, Edges, and Graphs
- Graph construction API (`@graph.node`, `builder.add_edge`, etc.)
- Types of edges: conditional, cyclical, default
- LangChain Expression Language (LCEL) in LangGraph
- Graph state management and immutability

Module 3: Building with LangGraph

- Creating simple linear graphs (e.g., sequential tasks)
- Conditional flows and branching logic
- Implementing loops and retries in LangGraph
- Asynchronous and parallel execution in LangGraph
- Error handling and fallback mechanisms in graphs

Module 4: Integrating LLMs in LangGraph

- Using LLMs as nodes in graphs
- Prompting strategies within graph workflows
- Combining tools (search, calculator, API) in LangGraph
- Using memory and context in LangGraph flows
- Tool calling and function-based interactions with OpenAI

Module 5: LangGraph for Multi-Agent Systems

- What are multi-agent systems in LangGraph?
- Creating agents as nodes with custom behavior
- Managing inter-agent communication and state
- Building conversational loops between agents
- Use cases: Debate bots, planner-executor systems, AI orchestration

Module 6: State and Memory Management

- Designing a state object (`dict`, `TypedDict`, `pydantic`)
- Passing and updating state through the graph
- Shared memory vs isolated memory per node
- Storing and retrieving memory from vector stores

Module 7: Advanced Patterns and Architectures

- Planner-executor pattern in LangGraph
- Task decomposition workflows
- Autonomous agents with feedback loops

- Implementing dynamic workflows (graph mutation at runtime)
- LangGraph with function-calling and OpenAI tools

Module 8: Integration with External Systems

- LangGraph + LangChain + tools (retrievers, APIs)
- Calling external APIs within graph nodes
- Working with vector databases (FAISS, Chroma, Weaviate, Pinecone)
- Integration with frontends (Streamlit, FastAPI, Gradio)
- Using LangGraph in cloud environments (serverless, containers)

Module 9: Observability, Testing & Debugging

- LangSmith integration for tracing and debugging graphs
- Logging and visualizing graph executions
- Testing individual nodes and graph branches
- Input/output validation with `pydantic`

Module 10: Deployment & Best Practices

- Packaging LangGraph apps as APIs (LangServe, FastAPI)
- Deployment to cloud (Vercel, AWS Lambda, etc.)
- Security practices (e.g., prompt injection protection)
- Managing costs and rate-limits with LangGraph
- Scalability and parallelism in production graph

==================================

END

===================================