

Intro:

- Hello, im valentina susarret. I am a junior studying cs here at uic and glad to be your host for this workshop about langchain and ai agents.
- Explain ai agents and langchain
- Explain goals for the workshop
- Go over the set up process (get link on discord)
 - Download python, langchain, ollama, and code files

Code inspection #1 - basic stateless agent:

- Open up one of the stateless agent ipynb scripts (summarizer agent)
- Explain the basic things needed (ollama extension, prompt template)
- Explain the use cases for these extremely simplistic agents
 - And how traditional code can make them “smarter”
 - Changing the prompt if the inputted text reaches a specific sentence count

Code inspection #2 - stateful agent (bufferMem)

- Open up chatbot agent with bufferMem attached.
- Show how the chatbot now “remembers” messages that were sent back and forth.
- Explain what’s actually happening under the hood.
 - Every time we prompt the agent, all of the previous messages are being sent together with our newest one.
 - Gives the illusion that the llm is remembering various facts about us.
 - Explain the issues (can lead to slower and worse outputs)

Code inspection #3 - stateful agent (summaryMem)

- Open up chatbot agent with summaryMem attached.
- This time the agent is remembering prior prompts by first summarizing all previous messages sent. The summary then is put into the prompt window with the newest one.
- This can also have downsides such as there being too many prior messages to get an indepth summary of.

Code review #4 - agentic pipeline

- Show results of the business competitors demo.
 - Explain the potential of the project if they were to be more

Concept explanation of RAG - no code

- Explain what rag is in simple terms and how it’s used to power chatGPT and other genai programs.
- Issues with rag include data management, data quality, and data scalability.