Coffee Break @ 9:30 am CT — 15 mins

2 Coffee Breaks < AM — 10:00 am / 9:30 am, PM — 2:00 pm > 15 mins

Lunch — 12:00 pm CT — 1 hr

Yes = +1, No = +2

"Github Copilot" → AI Assistant (Coding)
→ Coding Assistant
→ Repo & Management Code

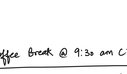
Generative AI Application →

FM | LLM (Foundation Model, Large Language Model)

LLM is a subset of FM
↳ Language (Content – Text, Image, video)

FM is pretrained Model which trained on Internet scale of data → RAW Data (Text, files, video, audio) Multi-Model Data

ML Model (Traditional) steps



FM / LLMs
→ Pretrained Model
→ Trained on Internet Scale of data (Raw Data)
→ Providers → Meta, Anthropic, Amazon, AI21 (Llama) (Claude) (Nova) (Jurassic)
→ OPENAI (GPT) fine-tune
→ "General Domain"

LLMs

TRANSFORMER ARCHI → 2014

Claude → # of tokens

Tokens = [Input + output]
50 250 = 300

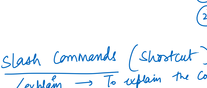
& → [ChatGPT] → (Reply)
Input (Translate "I love cats" into French) Context

[Translate | "I" | love | cats | into | French | .]
Tokens → numerical Rep → vector → vector DB

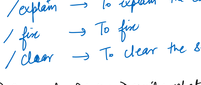
Context

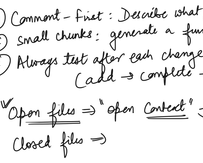
Translate - (I love cats) into French

Prompt (Instructions)



LLM → Code, Text, summary, conversation





Github Copilot Chat → Chat window
→ inline chat ⇒
1. Shortcut → Ctrl + I
2. Right click on the selected lines/cursor → open inline chat

Slash Commands (Shortcut)
/explain → To explain the Code
/fix → To fix
/clear → To clear the session chat in chat window

1. Comment-first: Describe what you want
2. Small chunks: generate a function at a time
3. Always test after each change (add → complete → filter → repeat)

Open files ⇒ "open Context" → inline chat / → Copilot chat
Closed files ⇒

Day 2 – Agenda

1. Prompt Engineering – Best Practices
   → Debug-style prompting
   → Common Mistakes
   Hands-on ⇒ Refactor the Project
2. Core Copilot use cases:
   – Code generation (functions, classes, API)
   – Refactoring / Modernization
   → Documentation (docstrings, ReadME)
   → Unit Test & Integration Tests
3. Advance Copilot features
   → Code Review
   → Copilot with frameworks (React, Node.js, Java, Python)
   → Multi-file reasoning & context linking
   → Copilot for Data-related tasks (Python, SQL)

Hands-on
- ✓ Code Review
- ✓ Create CRUD functionality
- ✓ Build unit tests & Integration tests
- ✓ Debug & Improve Copilot-generated Code

4. Extending Copilot with MCP

MS Learn