Day 2

Lab & Course Material

① Microsoft Learn M.C.

Recap on Day 1 - "GitHub Copilot ↳ Models"

⇒ "AI-powered Code Assistant"

LLMs (Models) - very large models which are pretrained on large volumes of data (Internet scale of data)

Architecture = TRANSFORMER (Deep Learning)

GPT = Generative Pre-trained Transformer

① VSC - Extensions: GitHub Copilot + GitHub Copilot Chat

GitHub Copilot License

Free (30 days trial) → limited access on models

Pro ($10/month)

Business/Enterprise

② Configure GitHub Copilot = Preferences GitHub M.C. + VSC

③ Automate Code Suggestions

✓1) Inline chat → ctrl + I

2) Slash commands

✓3) Copilot chat

M365 Copilot License

Agents / Declarative Agents

VSC ↳ Teams Toolkit

Copilot Studio

Day 2

⇒ Microsoft 365 Copilot ?

- Instructions ⇒ Natural-language prompts
- M365 and integrate with M365 apps
- Reference LLM + own org data (Microsoft Graph)

Key Capabilities

1. Content Creation
2. Data Insights
3. Collaboration & Meetings
4. Copilot chat ⇒ conversational assistant.

✓ TRANSFORMER → (LLMs) ⇒ Natural-language prompts

Core Components

① Input embeddings

Input → Tokens → vectors (numerical representation) → positional info

→ [Input (vec) tokens]

② Encoder ③ Decoder ⑤ Output # tokens

Context

Models ⇒ Reasoning (Contexts)

✓ Claude (very fast / very creative)

Agent = Instruction + Configuration

Agents & Declarative Agents

→ Agents extend M365 Copilot with task-specific intelligence

Context Knowledge

→ Declarative Agents → No-code / low-code way to define what an agent knows, tells at behaves, what actions it can take.

'Claude'

'Agents' in Microsoft 365 Copilot

- Specialized AI Agent/assistant → focusing business tasks
- Role / task specific Copilot
- What Agents can do:
  - ⇒ Q&A / FAQ using (enterprise data/knowledge)
  - ⇒ Perform multi-step tasks (research → sort → respond, Q → route → ...)
  - → Call tools & actions (Graph, connectors, APIs)
  - → Follow business rules & guardrails

When they live (Agents)

1. Copilot Chat
2. Teams
3. Word / PPT / Excel (Context)
4. M365 Copilot app

Ex: HR Policy Assistant, Finance Close Assistant, IT Helpdesk Assistant, Sales Proposal Agent

Declarative Agents

↳ Agents defined using Configuration (no code) (Declarative definition)

Key difference: Agents vs (Declarative Agents)

Dev style	Code + Config
Target audience	Pro dev
Tools	API, SDK, Apps
Hosting	Ext. platform
Spec-driven	Configuration
IT Admins, Power users	
MS-native connectors	
Native to M365 Copilot	
very fast	
Claw	

What you declare in Declarative Agent ?

① Agent Identity
① Name, Purpose, Scope (can & cannot)

② Knowledge Sources
① SharePoint sites
② OneDrive folders
③ MS Graph data
④ Enterprise connectors

③ Instructions (Behavior)
1) Tone (formal / friendly)
2) Boundaries
3) Compliance rules

④ Actions
- Calendar
- Tasks
- Records
- Trigger workflows

① Create instructions for the agent
② Publish Agent → M365 Copilot Chat
③ Update instructions / add knowledge
④ Add Actions / Connectors / APIs

Model → New

Model → fine-tune → Parameters → Datasets → Train the model

LLM + KB

User → Input (Text or Voice) → Business Logic → Model → Response

Prompt + Context

Context

KB

LLM (Brain) Knowledge

KB: SP, OD, MS Graph

✓ Type Spec ?

↳ open-source framework / language - Microsoft.

↳ simplifies the development for Agents.

↳ Microsoft 365 Copilot = Domain-specific language (DSL)

✓→ Declarative agents (instructions, instructions, conversation starters, knowledge data, etc)

Why ?

✓ Manageable ← yes

Declarative agents = instruction + knowledge + actions

Typespec = safe, clean way to author that experience & generate the manifest

'Microsoft Graph Connector'

'Graph Connector'

↳ Bring External Data into M365

Copilot → Agents

Microsoft Graph → Internal → 'Microsoft 365 data'

- Users
- Teams chat
- Files (OneDrive / SP)
- Email
- → SP Content
- → Calendar events