1:00 pm CST ⟶ 2:00 pm CST (15 mins)

WHAT — WHY — HOW

① Visual Studio Code ✓ ⟶ IDE

② github A/c (Personal) → 30 days

↳ Yes = +1 , No = +2

Copilot → Microsoft Product

↳ Microsoft 365 office ⟶ Word, Excel, PPT, (Outlook)

↳ Power Apps

↳ Development → ⟶ Amazon Q Developer

"GitHub" "Copilot" ✓ OPENAI

Interact with Copilot

① Inline suggestions = working on repetitive tasks

② Command palette = A quick access to the various functions of GitHub Copilot.

③ Copilot Chat = Interactive feature → ask Qs, request for code, explanation.

LLM

④ Inline chat = Context-specific conversation with Copilot chat

ⓐ place your cursor where you want the assistance.

ⓑ keyboard shortcut = Ctrl+I / Cmd+I (Mac)

ⓒ Ask Qs, code suggestion, etc.

↳ slash commands.

/explain ⇒ Explanation of the selected code

/suggest → offer the code suggestion - selected code

→ /tests → Generate Unit tests.

/comment → convert the comments into code snippets.

⑤ Multiple suggestions ⇒ Adv

⑥ Explanations →

⑦ Generate Unit Tests

"Prompt Engineering"

↳ Process of crafting clear instructions to guide AI system, to generate context-oriented code.

PRINCIPLES: (4 Ss)

- Single : single, well-defined task/Qs.
- Specific = Instructions are detailed.
- Short = Concise, to the point.
- Surround = Utilize descriptive file names, keep the related files open.

Best Practices

① Provide enough clarity.

② Provide enough context with details. (Surround)

③ Provide examples for learning.

? How Copilot learns from the prompts? Copilot

LLM

① Zero-shot learning - No Example.

② one-shot learning ↳ single example.

③ Few-shot learning ↳ several examples ✓ ↳ fine tune AI Models.

GenAI

↳ FMs

↳ LLMs

Human like language.