

# MS Learn

## MS Learn Q&A SumUp:

### Session 1:

1. What functionality isn't supported in GitHub Copilot for Individuals? \*

✓ **VPN Proxy support via self-signed certificates**

✗ **Offers multi-line function suggestions**

✗ **Editor integration**

✗ **Blocks suggestions matching public code**

2. What percentage of developers said that GitHub Copilot helps them code faster? \*

✗ **70%**

✗ **83%**

✗ **] 65%**

✓ **90%**

3. After you enforced your GitHub Copilot for Business policy, where do you first navigate to in order to enable Copilot for Business for all current and future users? \*

✗ **Policies**

✓ **Your organizations in your profile dropdown menu**

× **Settings in your profile dropdown menu**

× **Selected teams/users**

## **Session 2:**

1. What's GitHub Copilot? \*

× **A platform for code repository.**

× **A model powered by machine learning.**

✓ **An assistant for coding, powered by OpenAI.**

× **A service for web hosting.**

2. What role does prompting play in utilizing GitHub Copilot effectively? \*

× **Generates instant bug fixes.**

✓ **Enhances code suggestions quality.**

× **Automates the coding process entirely.**

× **Implements real-time collaboration.**

3. Which of the following is a principle of the 4 S's of prompt engineering? \*

× **Summarize code objectives concisely.**

✓ **Specify instructions explicitly and in detail.**

× **Streamline processes for efficient code suggestions.**

× **Simplify coding languages for universal understanding.**

4. How does GitHub Copilot handle personal data? \*

✗ It saves all personal data for future references.

✗ It shares personal data with other users for collaborative projects.

✗ It encrypts personal data.

✓ It actively filters out personal data to protect user privacy.

5. What is LoRA in the context of LLMs fine-tuning?

✓ A method that adds trainable elements to each layer of the pretrained model without a complete overhaul.

✗ A technology optimizing communication between different coding languages.

✗ A specialized software library enhancing Copilot's performance.

✗ A new programming paradigm supported exclusively by Copilot.

6. How does Copilot use the context to provide code suggestions? \*

✗ It considers only the prompt text you provide.

✗ It considers the file type but not the content of the file.

✓ It considers the surrounding code, file type, and content of parallel open tabs in the code editor.

✗ It randomly selects context from the internet.

7. Which of the following helps to improve prompt effectiveness in GitHub Copilot? \*

✓ Providing detailed contextual information with clarity.

✗ Making the prompt as general as possible.

✗ **Keeping the prompt lengthy and detailed.**

✗ **Avoiding examples in the prompt to not restrict Copilot's creativity.**

## Session 3:

1. Which of these advanced features aren't available in GitHub Copilot Enterprise but were available in GitHub Copilot Business? \*

✗ **Copilot Chat Customized to your Codebase**

✗ **Copilot Pull Request Summaries**

✗ **Copilot Documentation Search and Summaries using Docsets**

✓ **None of the above**

2. How does Copilot use an organization's codebase and internal knowledge to enhance productivity and collaboration? \*

✗ **By providing code suggestions based on open-source libraries only**

✓ **By tailoring coding assistance, answering questions, and suggesting code aligned with the organization's standards and best practices**

✗ **By suggesting code without considering the project context**

✗ **By randomly generating code snippets**

3. Describe the purpose and benefits of Copilot's pull request summaries in GitHub Copilot Enterprise. \*

✗ **They aim to provide a detailed history of code changes in pull requests**

✓ **They assist in automatically generating concise overviews of pull requests based on code changes,**

✗ enhancing understanding and accelerating review processes

✗ They track developers' activity within a pull request

✗ None of the above

4. How can organizations manage and utilize docsets within Copilot Enterprise to tailor code suggestions and improve development workflows? \*

✓ By using docsets to create custom collections of internal code and documentation

✗ By using docsets to automatically generate code snippets

✗ By using docsets to track developers' activity within a project

✗ By using docsets to enforce coding standards

## Session 4

1. How does GitHub Copilot work? \*

✓ GitHub Copilot uses prompts, natural language text, that you type, and it provides suggestions based on what you type.

✗ GitHub Copilot uses lights, that you type, and it provides suggestions based on what you type.

✗ GitHub Copilot uses radio language, that you type, and it provides suggestions based on what you type.

2. Is GitHub Copilot free? \*

✗ Yes, it's free for everyone.

✓ No, it's a service you can sign up for that's free for students to use but currently costs 10 dollars per month.

✗ **It's not free, even if you're a student or a teacher.**

3. How can you accept GitHub Copilot's suggestions? \*

✓ **Press the "Tab" key.**

✗ **Press "FI" key.**

✗ **Press "F4" key.**

4. Identify which statement is valid and select the correct answer: \*

✗ **A prompt, which is our output, is a collection of songs that tells our copilot what to generate.**

✓ **A prompt, which is our input, is a collection of instructions or guidelines that tell our copilot what to generate.**

✗ **A prompt, which is our document, is a collection of laptops that tells our Copilot what to generate.**

5. On what depends on the quality of the output from GitHub Copilot? \*

✗ **Your code editor.**

✗ **How well your extensions were installed.**

✓ **How well you crafted your prompt.**

## Session 5

1. What is ghost text in GitHub Copilot? \*

✓ **Ghost text in GitHub Copilot are suggestions that appears in your text editor as you type.**

✗ **Ghost text in GitHub Copilot are options used when typing to provide suggestions.**

- ✗ **Ghost text in GitHub Copilot involves using prompts and natural language questions within your code or documentation.**

2. How do you access GitHub Copilot's inline chat? \*

- ✗ **Access the inline chat by clicking on the chat icon in the left sidebar of Visual Studio Code.**

✓ **Use Ctrl+i on Windows or Command+i on a mac to open the inline chat.**

- ✗ **Access the inline chat by using Alt+i on Windows or Option+i on a mac.**

3. What are slash commands used for in GitHub Copilot? \*

- ✗ **Slash commands are used to format your codebase according to best practices.**

- ✗ **Slash commands are used to debug code and detect security vulnerabilities within your projects.**

✓ **Slash commands are shortcuts to quickly solve common development tasks within the chat or inline pane.**

4. What are the benefits of using agents like '@terminal' or '@workspace' when interacting with GitHub Copilot? \*

✓ **Agents in Visual Studio Code help you ask questions within a specific context, allowing for more precise and relevant answers from GitHub Copilot.**

- ✗ **Agents help enforce a consistent code format based on best practices within Visual Studio Code for improved readability.**

- ✗ **Agents provide extra security features for detecting vulnerabilities and intrusions within Visual Studio Code projects.**

5. What are the benefits of using implicit prompts with slash commands in inline chat for fixing code issues with GitHub Copilot? \*

- ✗ **Implicit prompts help enforce a consistent naming convention and syntax based on best practices within Visual Studio Code projects for improved**

**readability.**

✓ **Implicit prompts help get better responses from GitHub Copilot without writing longer prompts, making it easier to interact and fix code issues.**

× **Implicit prompts help detect security vulnerabilities and potential malicious activities within Visual Studio Code projects for increased safety.**