Welcome to Cert007 - Your Ultimate IT Certification Partner

- Real Exam Questions
- Free Updates
- Expert Support
- Instant Access
- Money-Back Guarantee

Visit us at **https://www.cert007.com/** for more information

Exam : **GitHub Copilot**

Title : GitHub Copilot Certification Exam

Version : DEMO

1.Which of the following describes role prompting?
A. Describing in your prompt what your role is to get a better suggestion
B. Tell GitHub Copilot in what tone of voice it should respond
C. Prompt GitHub Copilot to explain what was the role of a suggestion
D. Giving GitHub Copilot multiple examples of the form of the data you want to use
Answer: A
Explanation:
Role prompting involves explicitly stating your role or the persona you want GitHub Copilot to adopt within your prompt. This helps Copilot provide more contextually relevant and accurate suggestions. By defining your role (e.g., "As a senior software engineer," "As a technical writer"), you guide Copilot to tailor its responses to align with the expertise and perspective associated with that role. This improves the quality and relevance of the generated code and explanations.
Reference: GitHub Copilot documentation on prompt engineering and best practices.

2.Which of the following scenarios best describes the intended use of GitHub Copilot Chat as a tool?
A. A complete replacement for developers generating code.
B. A productivity tool that provides suggestions, but relying on human judgment.
C. A solution for software development, requiring no additional input or oversight.
D. A tool solely designed for debugging and error correction.
Answer: B
Explanation:
GitHub Copilot Chat is designed to be a productivity enhancer, not a replacement for human developers. It provides suggestions and assists with coding tasks, but the final decision and validation always rest with the developer. Copilot Chat is meant to augment the developer's workflow, making it faster and more efficient, but it does not remove the need for human oversight and judgment.
Reference: GitHub Copilot official documentation on the tool's purpose and usage.

3.How long does GitHub retain Copilot data for Business and Enterprise? (Each correct answer presents part of the solution. Choose two.)
A. Prompts and Suggestions: Not retained
B. Prompts and Suggestions: Retained for 28 days
C. User Engagement Data: Kept for Two Years
D. User Engagement Data: Kept for One Year
Answer: B, C
Explanation:
For GitHub Copilot Business and Enterprise, prompts and suggestions are retained for 28 days to provide context and improve the service. User engagement data, which includes usage patterns and interactions, is kept for two years. This data retention policy is designed to balance service improvement with user privacy.
Reference: GitHub Copilot documentation on data privacy and retention policies for Business and Enterprise plans.

4.What types of prompts or code snippets might be flagged by the GitHub Copilot toxicity filter? (Each correct answer presents part of the solution. Choose two.)

A. Hate speech or discriminatory language (e.g., racial slurs, offensive stereotypes)
B. Sexually suggestive or explicit content
C. Code that contains logical errors or produces unexpected results
D. Code comments containing strong opinions or criticisms
Answer: A, B
Explanation:
GitHub Copilot includes a toxicity filter to prevent the generation of harmful or inappropriate content. This filter flags prompts or code snippets that contain hate speech, discriminatory language, or sexually suggestive or explicit content. This ensures a safe and respectful coding environment.
Reference: GitHub Copilot documentation on safety and content filtering.

5.What is a benefit of using custom models in GitHub Copilot?
A. Responses are faster to produce and appear sooner
B. Responses use practices and patterns in your repositories
C. Responses use the organization's LLM engine
D. Responses are guaranteed to be correct
Answer: B
Explanation:
Custom models in GitHub Copilot allow the tool to learn from the specific code patterns and practices within your repositories. This results in suggestions that are more aligned with your organization's coding standards and conventions, improving the relevance and accuracy of the generated code.
Reference: GitHub Copilot Enterprise documentation on custom models.