Product Questions: 120
Version: 8.2

Question: 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.

Question: 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.

Question: 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 andEnterprise plans.

Question: 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.

Question: 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.

Question: 6

How does GitHub Copilot Chat ensure that a function works correctly?

A. By suggesting assertions based on the code's context and semantics.
B. By automatically writing all the tests for the function.
C. By writing the implementation code for the function.
D. By executing the test cases to validate the correctness of the code.

Answer: A

Explanation:

GitHub Copilot Chat can suggest assertions based on the code's context and semantics to help developers verify the correctness of their functions. These assertions serve as checks that the function behaves as expected under various conditions.
Reference: GitHub Copilot documentation on testing and code verification.

Question: 7

What is a key consideration when relying on GitHub Copilot Chat's explanations of code functionality and proposed improvements?

A. The explanations are dynamically updated based on user feedback.
B. Reviewing and validating the generated output for accuracy and completeness.
C. GitHub Copilot Chat uses a static database for generating explanations.
D. The explanations are primarily derived from user-provided documentation.

Answer: B

Explanation:

While GitHub Copilot Chat can provide helpful explanations and suggestions, it's crucial to review and validate the generated output. Copilot's suggestions are based on its training data, and they may not always be perfectly accurate or complete. Human judgment is essential to ensure the quality and correctness of the code.
Reference: GitHub Copilot best practices and usage guidelines.

Question: 8

How can GitHub Copilot assist with code refactoring tasks?

A. GitHub Copilot can fix syntax errors without user input.
B. GitHub Copilot can automatically rewrite code to follow best practices.
C. GitHub Copilot can suggest refactoring improvements for better code quality.
D. GitHub Copilot can remove unnecessary files from the project directory.

Answer: C

Explanation:

GitHub Copilot can analyze existing code and suggest refactoring improvements to enhance code quality, readability, and maintainability. It can propose changes to improve code structure, reduce complexity, and follow best practices.
Reference: GitHub Copilot documentation on code refactoring assistance.

Question: 9

What is the process behind identifying public code matches when using a public code filter enabled in GitHub Copilot?

A. Running code suggestions through filters designed to detect public code
B. Comparing suggestions against public code using machine learning.
C. Analyzing the context and structure of the code being written
D. Reviewing the user's browsing history to identify public repositories

Answer: A

Explanation:

When the public code filter is enabled, GitHub Copilot runs code suggestions through filters designed

to detect matches with publicly available code. This helps prevent the generation of code that might infringe on copyright or licensing agreements.
Reference: GitHub Copilot documentation on public code filtering and licensing.

Question: 10

Which of the following prompts can be used to guide GitHub Copilot Chat in refactoring code for quality improvements? (Each correct answer presents part of the solution. Choose two.)

A. "Show me how to improve the readability of this function."
B. "Suggest ways to enhance the maintainability of this code segment."
C. "Refactor my application to meet the latest coding standards."
D. "Predict future coding trends and update my codebase accordingly."

Answer: A,B

Explanation:

Effective prompts for refactoring include requests that focus on specific quality improvements, such as readability and maintainability. These prompts guide GitHub Copilot to provide relevant and actionable suggestions.
Reference: GitHub Copilot prompt engineering best practices.

Question: 11

What method can a developer use to generate sample data with GitHub Copilot? (Each correct answer presents part of the solution. Choose two.)

A. Utilizing GitHub Copilot's ability to create fictitious information from patterns in training data.
B. Leveraging GitHub Copilot's ability to independently initiate and manage data storage services.
C. Utilize GitHub Copilot's capability to directly access and use databases to create sample data.
D. Leveraging GitHub Copilot's suggestions to create data based on API documentation in the repository.

Answer: A,D

Explanation:

GitHub Copilot can generate sample data by creating fictitious information based on patterns in its training data and by using suggestions based on API documentation within the repository.
Reference: GitHub Copilot documentation on data generation assistance.

Question: 12

What are the potential risks associated with relying heavily on code generated from GitHub Copilot? (Each correct answer presents part of the solution. Choose two.)

A. GitHub Copilot may introduce security vulnerabilities by suggesting code with known exploits.
B. GitHub Copilot may decrease developer velocity by requiring too much time in prompt engineering.
C. GitHub Copilot's suggestions may not always reflect best practices or the latest coding standards.
D. GitHub Copilot may increase development lead time by providing irrelevant suggestions.

Answer: A,C

Explanation:

Heavy reliance on GitHub Copilot can introduce security vulnerabilities if the generated code contains known exploits. Additionally, Copilot's suggestions may not always align with best practices or the latest standards, requiring careful review and validation.
Reference: GitHub Copilot best practices and risk management.

Question: 13

What is the correct way to access the audit log events for GitHub Copilot Business?

A. Navigate to the Security tab in the organization's GitHub settings
B. Navigate to the Insights tab in the repository settings
C. Use the Audit log section in the organization's GitHub settings
D. Use the Code tab in the GitHub repository

Answer: C

Explanation:

Audit log events for GitHub Copilot Business can be accessed through the Audit log section within the organization's GitHub settings. This log provides a record of activities related to Copilot usage and configuration.
Reference: GitHub Copilot Business documentation on audit logs.

Question: 14

How can you improve the context used by GitHub Copilot? (Each correct answer presents part of the solution. Choose two.)

A. By opening the relevant tabs in your IDE
B. By adding relevant code snippets to your prompt
C. By adding the important files to your .gitconfig

D. By adding the full file paths to your prompt of important files

Answer: A,B

Explanation:

Improving the context for GitHub Copilot involves opening relevant files in your IDE to provide immediate context and adding relevant code snippets directly to your prompts to give Copilot specific examples and information.
Reference: GitHub Copilot prompt engineering and context management.

Question: 15

How can GitHub Copilot assist developers during the requirements analysis phase of the Software Development Life Cycle (SDLC)?

A. By automatically generating detailed requirements documents.
B. By providing templates and code snippets that help in documenting requirements.
C. By identifying and fixing potential requirement conflicts when using /help.
D. By managing stakeholder communication and meetings.

Answer: B

Explanation:

GitHub Copilot can assist during the requirements analysis phase by providing templates and code snippets that aid in documenting requirements. This helps streamline the process of capturing and organizing project requirements.
Reference: GitHub Copilot documentation on SDLC assistance.

Question: 16

How do you generate code suggestions with GitHub Copilot in the CLI?

A. Write code comments -> Press the suggestion shortcut -> Select the best suggestion from the list.
B. Use copilot suggest -> Write the command you want -> Select the best suggestion from the list.
C. Type out the code snippet -> Use the copilot refine command to enhance it -> Review the suggested command.
D. Describe the project's architecture -> Use the copilot generate command -> Accept the generated suggestion.

Answer: A

Explanation:

In the CLI, GitHub Copilot generates code suggestions by analyzing code comments. You write comments describing what you want, and Copilot provides relevant code suggestions. You then select the best suggestion from the list.
Reference: GitHub Copilot CLI documentation.

Question: 17

Where is the proxy service hosted?

A. Self hosted
B. Amazon Web Service
C. Microsoft Azure
D. Google Cloud Platform

Answer: C

Explanation:

The proxy service for GitHub Copilot is hosted on Microsoft Azure.
Reference: GitHub Copilot infrastructure and hosting information.

Question: 18

What configuration needs to be set to get help from Microsoft and GitHub protecting against IP infringement while using GitHub Copilot?

A. Suggestions matching public code to 'blocked'
B. Enforce blocking of MIT or GPL licensed code
C. You need to check code suggestions yourself before accepting
D. Enable GitHub Copilot license checking

Answer: A

Explanation:

To help protect against IP infringement, you need to configure GitHub Copilot to block suggestions that match public code. This ensures that the generated code is not directly copied from publicly available sources.
Reference: GitHub Copilot documentation on IP protection and code filtering.

Question: 19

Which principle emphasizes that AI systems should be understandable and provide clear information on how they work?

A. Fairness
B. Transparency
C. Inclusiveness
D. Accountability

Answer: B

Explanation:

The principle of transparency emphasizes that AI systems should be understandable and provide clear information about their operations. This ensures that users can understand how the AI arrives at its decisions and suggestions.
Reference: Microsoft's AI principles and ethical guidelines.

Question: 20

In what way can GitHub Copilot and GitHub Copilot Chat aid developers in modernizing applications?

A. GitHub Copilot can directly convert legacy applications into cloud-native architectures.
B. GitHub Copilot can suggest modern programming patterns based on your code.
C. GitHub Copilot can create and deploy full-stack applications based on a single query.
D. GitHub Copilot can refactor applications to align with upcoming standards.

Answer: B

Explanation:

GitHub Copilot and GitHub Copilot Chat are powerful AI-driven tools designed to assist developers by providing context-aware code suggestions and interactive support. Specifically, in the context of modernizing applications, GitHub Copilot excels at analyzing existing code and suggesting modern programming patterns, best practices, and syntax improvements that align with contemporary development standards. For example, it can recommend updates to outdated constructs, propose more efficient algorithms, or suggest frameworks and libraries that are widely used in modern application development.
Why not A?GitHub Copilot does not "directly convert" legacy applications into cloud-native architectures. It can assist by suggesting code changes or patterns that support such a transition, but it doesn’t autonomously perform the full conversion process, which involves architectural decisions and deployment steps beyond its scope.
Why not C?While GitHub Copilot can generate code snippets and even larger portions of an application, it cannot create and deploy full-stack applications from a single query. It requires developer input, refinement, and integration to achieve a complete, deployable solution.
Why not D?GitHub Copilot can assist with refactoring by suggesting improvements to existing code, but it doesn’t inherently "align with upcoming standards" in a predictive sense. Its suggestions are based on current best practices and the data it was trained on, not future standards that are yet to be