



"Please note that these files may not be up to date. However, the questions will help you understand the exam format and typical question patterns."

www.atmicnetworks.com

Warning: Keep connected with our support team for latest updates

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 and Enterprise 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 defined.

Thus, B is the most accurate and realistic way GitHub Copilot aids developers in modernizing applications, leveraging its ability to provide relevant, context-based suggestions to update and improve codebases.

Reference: GitHub Copilot documentation on application modernization.

Question: 21

How does GitHub Copilot Chat utilize its training data and external sources to generate responses when answering coding questions?

- A. It primarily relies on the model's training data to generate responses.
- B. It primarily uses search results from Bing to generate responses.
- C. It combines its training data set, code in user repositories, and external sources like Bing to generate responses.
- D. It uses user-provided documentation exclusively to generate responses.

Answer: C

Explanation:

GitHub Copilot Chat combines its training data, code from user repositories, and external sources like Bing to generate comprehensive and relevant responses to coding questions.

Reference: GitHub Copilot Chat documentation on data sources.

Question: 22

How does GitHub Copilot Chat help in understanding the existing codebase?

- A. By running code linters and formatters.
- B. By providing visual diagrams of the code structure.
- C. By answering questions about the code and generating explanations.
- D. By automatically refactoring code to improve readability.

Answer: C

Explanation:

GitHub Copilot Chat helps in understanding existing codebases by answering questions about the code and generating explanations. This allows developers to quickly grasp the functionality and structure of unfamiliar code.

Reference: GitHub Copilot Chat documentation on code understanding.

Question: 23

GitHub Copilot in the Command Line Interface (CLI) can be used to configure the following settings: (Each correct answer presents part of the solution. Choose two.)

- A. The default execution confirmation
- B. Usage analytics
- C. The default editor
- D. GitHub CLI subcommands

Answer: A, B

Explanation:

GitHub Copilot in the CLI allows configuration of settings such as the default execution confirmation and usage analytics. These settings help tailor the CLI experience to the user's preferences.

Reference: GitHub Copilot CLI configuration documentation.

Question: 24

How does the /tests slash command assist developers?

- A. Constructs detailed test documentation.
- B. Creates unit tests for the selected code.
- C. Integrates with external testing frameworks.
- D. Executes test cases to find issues with the code.

Answer: B

Explanation:

The /tests slash command in GitHub Copilot Chat creates unit tests for the selected code, helping developers ensure the functionality and reliability of their code.

Reference: GitHub Copilot Chat command documentation.

Question: 25

What GitHub Copilot pricing plan gives you access to your company's knowledge bases?

- A. GitHub Copilot Individual
- B. GitHub Copilot Business
- C. GitHub Copilot Enterprise
- D. GitHub Copilot Professional

Answer: C

Explanation:

GitHub Copilot Enterprise provides access to your company's knowledge bases, enabling the tool to provide contextually relevant suggestions based on your organization's specific documentation and code.

Reference: GitHub Copilot Enterprise pricing and features.

Question: 26

How can the concept of fairness be integrated into the process of operating an AI tool?

- A. Focusing on accessibility will ensure fairness.
- B. Focusing on collecting large datasets for training will ensure fairness.
- C. Regularly monitoring the AI tool's performance will ensure fairness in its outputs.
- D. Training AI data and algorithms to be free from biases will ensure fairness.

Answer: D

Explanation:

Fairness in AI tools is achieved by training the data and algorithms to be free from biases. This ensures that the tool treats all users equitably and avoids discriminatory outcomes.

Reference: Microsoft's AI principles and fairness guidelines.

Question: 27

How can you use GitHub Copilot to get inline suggestions for refactoring your code? (Select two.)

- A. By adding comments to your code and triggering a suggestion.
- B. By highlighting the code you want to fix, right-clicking, and selecting "Fix using GitHub Copilot."
- C. By running the gh copilot fix command.
- D. By using the "/fix" command in GitHub Copilot in-line chat.
- E. By highlighting the code you want to fix, right-clicking, and selecting "Refactor using GitHub Copilot."

Answer: A, E

Explanation:

You can use GitHub Copilot for inline refactoring suggestions by adding comments to your code to trigger suggestions and by highlighting the code and selecting "Refactor using GitHub Copilot" from the context menu.

Reference: GitHub Copilot refactoring documentation.

Question: 28

What are the potential limitations of GitHub Copilot Chat? (Each correct answer presents part of the solution. Choose two.)

- A. Limited training data
- B. Ability to handle complex code structures
- C. No biases in code suggestions
- D. Extensive support for all programming languages

Answer: A, D

Explanation:

GitHub Copilot Chat has limitations such as limited training data, which can affect the accuracy of its suggestions, and it does not provide extensive support for all programming languages.

Reference: GitHub Copilot Chat limitations documentation.

Question: 29

What are the effects of content exclusions? (Each correct answer presents part of the solution. Choose two.)

- A. The excluded content is not directly available to GitHub Copilot to use as context.
- B. GitHub Copilot suggestions are no longer available in the excluded files.
- C. The excluded content is no longer used while debugging the code.
- D. The IDE will not count coding suggestions in the excluded content.

Answer: A, B

Explanation:

Content exclusions prevent GitHub Copilot from using the excluded content as context and stop suggestions from being generated in those files.

Reference: GitHub Copilot content exclusion documentation.

Question: 30

What role does chat history play in GitHub Copilot's code suggestions?

- A. Chat history is used to train the GitHub Copilot model in real-time.
- B. Chat history provides context to GitHub Copilot, improving the relevance and accuracy of its code suggestions.
- C. Chat history is stored and shared with other users to enhance collaboration.
- D. Chat history is irrelevant to GitHub Copilot and does not affect its functionality.

Answer: B

Explanation:

Chat history provides valuable context to GitHub Copilot, helping it generate more relevant and accurate code suggestions based on previous interactions and conversations.

Reference: GitHub Copilot Chat history documentation.

Question: 31

How can GitHub Copilot facilitate a smoother learning experience when diving into a new programming language? (Each correct answer presents part of the solution. Choose two.)

- A. GitHub Copilot Chat can provide guidance and support for common coding tasks and challenges in the targeted programming language.
- B. GitHub Copilot's /understand command will help GitHub Copilot to understand code written in a targeted programming language.
- C. GitHub Copilot can provide contextualized code suggestions and answer sources from an organization's documentation.
- D. GitHub Copilot can convert comments into code to grasp the syntax and nuances of a new programming language.

Answer: A, D

Explanation:

GitHub Copilot helps with learning new languages by providing guidance on common tasks and by converting comments into code, allowing developers to see syntax in action.

Reference: GitHub Copilot language learning documentation.

Question: 32

What is the correct way to exclude specific files from being used by GitHub Copilot Business during code suggestions?

- A. Modify the .gitignore file to include the specific files.
- B. Add the specific files to a copilot.ignore file.
- C. Use the GitHub Copilot settings in the user interface to exclude files.
- D. Rename the files to include the suffix _no_copilot.

Answer: C

Explanation:

The correct way to exclude files is through the GitHub Copilot settings in the user interface, which allows administrators to specify files and directories to be ignored.

Reference: GitHub Copilot Business content exclusion documentation.

Question: 33

What method can be used to interact with GitHub Copilot?

- A. By using a properly configured GitHub CLI
- B. By using chat capabilities in NeoVim
- C. From a watch window in an IDE debug session
- D. From a web browser at <https://github.copilot.com>

Answer: B

Explanation:

GitHub Copilot is an AI-powered code completion tool that integrates directly into supported Integrated Development Environments (IDEs) and code editors, such as Visual Studio Code, JetBrains IDEs, NeoVim, and others. Developers interact with it through their coding environment, where it provides real-time code suggestions, autocompletions, and (in some cases) chat-like capabilities via extensions or plugins (e.g., GitHub Copilot Chat in supported editors).

Evaluation of Options:

- A. By using a properly configured GitHub CLI

The GitHub CLI (Command Line Interface) is a tool for interacting with GitHub repositories and workflows from the terminal, but it is not a method for interacting with GitHub Copilot. Copilot operates within code editors/IDEs, not through the CLI. Incorrect.

- B. By using chat capabilities in NeoVim

This is partially correct. GitHub Copilot can be used in NeoVim with the appropriate plugin (e.g., the Copilot.vim plugin), and GitHub Copilot Chat—a feature that allows conversational interaction—may also be available depending on the setup and version. However, "chat capabilities in NeoVim" alone is not the primary or standard way to describe Copilot interaction, as it's more about code suggestions than chat. This is the closest option but not perfectly precise. Partially correct.

- C. From a watch window in an IDE debug session

The "watch window" in an IDE is used during debugging to monitor variable values, not to interact with GitHub Copilot. Copilot provides suggestions while coding, not specifically in debug sessions or watch windows. Incorrect.

- D. From a web browser at <https://github.copilot.com>

There is no such URL as "<https://github.copilot.com>" dedicated to interacting with GitHub Copilot. Copilot is accessed via GitHub's authentication and integrated into editors/IDEs, not through a standalone web browser interface. Information about Copilot is available on GitHub's official site (e.g.,

<https://github.com/features/copilot>), but interaction happens in the coding environment. Incorrect.

Reference: GitHub Copilot interaction documentation.

Question: 34

What practices enhance the quality of suggestions provided by GitHub Copilot? (Select three.)

- A. Clearly defining the problem or task
- B. Including personal information in the code comments

- C. Using meaningful variable names
- D. Providing examples of desired output
- E. Use a .gitignore file to exclude irrelevant files

Answer: A, C, D

Explanation:

The quality of Copilot's suggestions is enhanced by clearly defining the task, using meaningful variable names, and providing examples of the desired output.

Reference: GitHub Copilot prompt engineering best practices.

Question: 35

Which Microsoft ethical AI principle is aimed at ensuring AI systems treat all people equally?

- A. Privacy and Security
- B. Fairness
- C. Reliability and Safety
- D. Inclusiveness

Answer: B

Explanation:

The principle of fairness ensures that AI systems treat all people equally and avoid discriminatory outcomes.

Reference: Microsoft's AI principles and fairness guidelines.

Question: 36

Which of the following is correct about GitHub Copilot Knowledge Bases?

- A. All repos are indexed
- B. Indexing is static
- C. It is an Enterprise feature
- D. All file types are indexed

Answer: C

Explanation:

GitHub Copilot Knowledge Bases is an Enterprise feature that allows the tool to use an organization's internal documentation and code for more accurate and relevant suggestions.

Reference: GitHub Copilot Enterprise Knowledge Base documentation.

Question: 37

Why is code reviewing still necessary when using GitHub Copilot to write tests?

- A. Because GitHub Copilot can cover all possible scenarios in your test cases.
- B. Because GitHub Copilot generates the best code possible for the test scenario.
- C. Because GitHub Copilot's generated test cases may not cover all possible scenarios.
- D. Because GitHub Copilot replaces the need for manual testing.

Answer: C

Explanation:

Code review is necessary because GitHub Copilot's generated test cases might not cover all possible scenarios, especially edge cases and complex interactions.

Reference: GitHub Copilot testing best practices.

Question: 38

A company is currently storing code in Bitbucket and would like to use GitHub Copilot. Which GitHub Copilot plan will be most cost effective to allow them to manage users with their Identity Provider (e.g. Okta)?

- A. GitHub Copilot Business for non-GHE customers
- B. GitHub Copilot Individual
- C. GitHub Copilot Enterprise
- D. GitHub Copilot Teams

Answer: C

Explanation:

GitHub Copilot Enterprise is the most cost-effective plan for managing users with an Identity Provider like Okta, as it provides enterprise-level features and integration.

Reference: GitHub Copilot pricing and features.

Question: 39

How long does it take content exclusion to add or be updated?

- A. Up to 30 minutes
- B. 45 - 60 minutes
- C. 60 - 90 minutes
- D. 24 hours

Answer: A

Explanation:

Content exclusions typically take up to 30 minutes to be added or updated.

Reference: GitHub Copilot content exclusion documentation.

Question: 40

How is GitHub Copilot Individual billed? (Each correct answer presents part of the solution. Choose two.)

- A. Monthly as a subscription
- B. Annually as a subscription
- C. Monthly, as a metered service based on actual consumption
- D. Free (not billed) for all open source projects

Answer: A, B

Explanation:

GitHub Copilot Individual is billed as a monthly or annual subscription.
Reference: GitHub Copilot Individual pricing.

Question: 41

What is the best way to share feedback about GitHub Copilot Chat when using it on GitHub Mobile?

- A. The Settings menu in the GitHub Mobile app.
- B. The feedback section on the GitHub website.
- C. Use the emojis in the Copilot Chat interface.
- D. By tweeting at GitHub's official X (previously known as Twitter) account.

Answer: C

Explanation:

The best way to share feedback on GitHub Mobile is using the emojis in the Copilot Chat interface, which allows for direct and immediate feedback.
Reference: GitHub Copilot Mobile documentation.

Question: 42

How can you get multiple suggestions from GitHub Copilot?

- A. By asking for multiple suggestions using comments in your code
- B. By opening the completions panel in your editor
- C. By using the inline chat functionality with the command 'multiple'
- D. By using @workspace in the chat window

Answer: B

Explanation:

You can get multiple suggestions by opening the completions panel in your editor, which displays alternative code suggestions.

Reference: GitHub Copilot usage documentation.

Question: 43

Select a strategy to increase the performance of GitHub Copilot Chat.

- A. Use a single GitHub Copilot Chat query to find resolutions for the collection of technical requirements
- B. Limit the number of concurrent users accessing GitHub Copilot Chat
- C. Apply prompt engineering techniques to be more specific
- D. Optimize the usage of memory-intensive operations within generated code

Answer: C

Explanation:

Applying prompt engineering techniques to be more specific is the best strategy to increase the performance and relevance of GitHub Copilot Chat's responses.

Reference: GitHub Copilot prompt engineering best practices.

Question: 44

What should developers consider when relying on GitHub Copilot for generating code that involves statistical analysis?

- A. GitHub Copilot can independently verify the statistical significance of results.
- B. GitHub Copilot's suggestions are based on statistical trends and may not always apply accurately to specific datasets.
- C. GitHub Copilot will automatically correct any statistical errors found in the user's initial code.
- D. GitHub Copilot can design new statistical methods that have not been previously documented.

Answer: B

Explanation:

Developers should consider that GitHub Copilot's suggestions are based on statistical trends and may NOT always be accurate for specific datasets, requiring careful validation.

Reference: GitHub Copilot data analysis limitations.

Question: 45

Which Copilot Individual features are available when using a supported extension for Visual Studio, VS Code, or JetBrains IDEs? (Each correct answer presents part of the solution. Choose two.)

- A. Code suggestions
- B. Chat
- C. Knowledge Base
- D. Pull Request Diff Analysis

Answer: A, B

Explanation:

GitHub Copilot Individual provides code suggestions and chat features when used with supported IDE extensions like Visual Studio, VS Code, and JetBrains IDEs.

Reference: GitHub Copilot Individual features documentation.

Question: 46

What are two techniques that can be used to improve prompts to GitHub Copilot? (Select two.)

- A. Provide specific success criteria
- B. Provide all information about the utilized files
- C. Provide insight on where to get the content from to get a response
- D. Provide links to supporting documentation

Answer: A, D

Explanation:

Improving prompts involves providing specific success criteria and including links to supporting documentation to give GitHub Copilot more context and direction.

Reference: GitHub Copilot prompt engineering best practices.

Question: 47

Identify the right use cases where GitHub Copilot Chat is most effective. (Each correct answer presents part of the solution. Choose two.)

- A. Create a technical requirement specification from the business requirement documentation
- B. Explain a legacy COBOL code and translate the code to another language like Python.
- C. Creation of a unit test scenario for newly developed Python code
- D. Creation of end-to-end performance testing scenarios for a web application

Answer: B, C

Explanation:

GitHub Copilot Chat is effective for explaining and translating legacy code and generating unit test scenarios for new code.

Reference: GitHub Copilot Chat use cases.

Question: 48

Which of the following GitHub Copilot Business related activities can be tracked using the organization audit logs?

- A. Accepted chat suggestions
- B. Code suggestions made by GitHub Copilot
- C. Changes to content exclusion settings
- D. Suggestions blocked by duplication detection filtering

Answer: C

Explanation:

Organization audit logs track changes to content exclusion settings, providing administrators with visibility into configuration changes.

Reference: GitHub Copilot Business audit logs.

Question: 49

What content can be configured to be excluded with content exclusions? (Each correct answer presents part of the solution. Choose three.)

- A. Lines in files
- B. Files
- C. Folders
- D. Repositories
- E. Gists

Answer: B, C, D

Explanation:

Content exclusions allow you to exclude files, folders, and repositories from being used by GitHub Copilot.

Reference: GitHub Copilot content exclusion documentation.

Question: 50

When can GitHub Copilot still use content that was excluded using content exclusion?

- A. If the contents of an excluded file are referenced in code that is not excluded, for example function calls.
- B. When the repository level settings allow overrides by the user.
- C. If the content exclusion was configured at the enterprise level, and is overwritten at the organization level.
- D. When the user prompts with @workspace.

Answer: A

Explanation:

GitHub Copilot can still use excluded content if it is referenced in code that is not excluded, such as function calls.

Reference: GitHub Copilot content exclusion documentation.

Question: 51

Which of the following is a risk associated with using AI?

- A. AI algorithms are incapable of perpetuating existing biases.
- B. AI systems can sometimes make decisions that are difficult to interpret.
- C. AI eliminates the need for data privacy regulations.
- D. AI replaces the need for developer opportunities in most fields.

Answer: B

Explanation:

A risk associated with AI is that its decisions can be difficult to interpret, leading to a lack of transparency and potential misunderstandings.

Reference: AI risk management documentation.

Question: 52

What is a limitation of content exclusions?

- A. Content exclusions can be worked around as it is only available for Git repositories.
- B. Content exclusions can only be configured by an enterprise administrator.
- C. Content exclusions are only available in the GitHub Copilot Individual plan.
- D. Repository administrators and organization owners cannot manage content exclusion settings.

Answer: A

Explanation:

A limitation is that content exclusions are only available for Git repositories, meaning they can be worked around if content is accessed through other means.

Reference: GitHub Copilot content exclusion documentation.

Question: 53

Why might a Generative AI (Gen AI) tool create inaccurate outputs?

- A. The Gen AI tool is overloaded with too many requests at once.
- B. The Gen AI tool is experiencing downtime and is not fully recovered.
- C. The Gen AI tool is programmed with a focus on creativity over factual accuracy.
- D. The training data might contain biases or inconsistencies.

Answer: D

Explanation:

Gen AI tools can produce inaccurate outputs if the training data contains biases or inconsistencies, which can lead to flawed or misleading results.

Reference: Generative AI limitations documentation.

Question: 54

An independent contractor develops applications for a variety of different customers. Assuming no concerns from their customers, which GitHub Copilot plan is best suited?

- A. GitHub Copilot Individual
- B. GitHub Copilot Business
- C. GitHub Copilot Business for non-GHE Customers
- D. GitHub Copilot Enterprise
- E. GitHub Copilot Teams

Answer: A

Explanation:

For an independent contractor, GitHub Copilot Individual is the most suitable and cost-effective plan.

Reference: GitHub Copilot pricing documentation.

Question: 55

What is a likely effect of GitHub Copilot being trained on commonly used code patterns?

- A. Suggest homogeneous solutions if provided a diverse data set.
- B. Suggest innovative coding solutions that are not yet popular.
- C. Suggest code snippets that reflect the most common practices in the training data.
- D. Suggest completely novel projects, while reducing time on a project.

Answer: C

Explanation:

GitHub Copilot is likely to suggest code snippets that reflect common practices in its training data, as it

learns from frequently used patterns.

Reference: GitHub Copilot training data documentation.

Question: 56

Which REST API endpoint is used to modify details about a GitHub Copilot Business subscription?

(Each correct answer presents part of the solution. Choose two.)

- A. Add teams to the Copilot subscription for an organization
- B. Upgrade or downgrade the subscription tier
- C. Migrate Copilot seat assignments between GitHub organizations
- D. Reassign Copilot seats based on GitHub repository size
- E. Remove teams from the Copilot subscription for an organization

Answer: A, E

Explanation:

The REST API endpoints are used to add and remove teams from the Copilot Business subscription within an organization.

Reference: GitHub Copilot Business API documentation.

Question: 57

What is the primary role of the /optimize slash command in Visual Studio?

- A. Automatically formats the code according to the selected style guide.
- B. Enhances the performance of the selected code by analyzing its runtime complexity.
- C. Summarizes your documentation into more maintainable and readable formats.
- D. Translates code into a more performant language.

Answer: B

Explanation:

The /optimize slash command in Visual Studio enhances the performance of the selected code by analyzing its runtime complexity and suggesting improvements.

Reference: GitHub Copilot Visual Studio commands.

Question: 58

What GitHub Copilot feature can be configured at the organization level to prevent GitHub Copilot suggesting publicly available code snippets?

- A. GitHub Copilot Chat in the IDE
- B. GitHub Copilot Chat in GitHub Mobile
- C. GitHub Copilot duplication detection filter
- D. GitHub Copilot access to Bing

Answer: C

Explanation:

The duplication detection filter can be configured at the organization level to prevent GitHub Copilot from suggesting publicly available code snippets.

Question: 59

What is zero-shot prompting?

- A. Giving as little context to GitHub Copilot as possible
- B. Telling GitHub Copilot it needs to show only the correct answer
- C. Only giving GitHub Copilot a question as a prompt and no examples
- D. Giving GitHub Copilot examples of the problem you want to solve
- E. Giving GitHub Copilot examples of the algorithm and outcome you want to use

Answer: C

Explanation:

Zero-shot prompting involves asking GitHub Copilot a question or giving a task without providing any examples. This relies on the model's pre-existing knowledge to generate a response.

Reference: GitHub Copilot prompt engineering documentation.

Question: 60

How can GitHub Copilot be limited when it comes to suggesting unit tests?

- A. GitHub Copilot can generate all types of unit tests, including those for edge cases and complex integration scenarios.
- B. GitHub Copilot primarily suggests basic unit tests that focus on core functionalities, often requiring additional input from developers for comprehensive coverage.
- C. GitHub Copilot can handle any complexity in code and automatically generate appropriate unit tests.
- D. GitHub Copilot's limitations in generating unit tests can vary based on the IDE version you are using.

Answer: B

Explanation:

GitHub Copilot often suggests basic unit tests and may not cover all edge cases or complex integration scenarios, requiring developers to supplement its suggestions.

Reference: GitHub Copilot testing limitations.

Question: 61

effectiveness of GitHub Copilot? (Each correct answer presents part of the solution. Choose two.) 1

1.

blog.yatricloud.com
blog.yatricloud.com

- A. The API can generate detailed reports on code quality improvements made by GitHub Copilot.
- B. The API can track the acceptance rate of code suggestions accepted and used in the organization.
- C. The API can refactor your code to improve productivity.
- D. The API can provide feedback on coding style and standards compliance.
- E. The API can provide Copilot Chat specific suggestions acceptance metrics.

Answer: B, E

Explanation:

The GitHub Copilot usage metrics API provides insights into the acceptance rate of code suggestions and Copilot Chat specific suggestions acceptance metrics, helping organizations evaluate its effectiveness.

Reference: GitHub Copilot usage metrics API documentation.

Question: 62

How does GitHub Copilot suggest code optimizations for improved performance?

- A. By analyzing the codebase and suggesting more efficient algorithms or data structures.
- B. By automatically rewriting the codebase to use more efficient code.
- C. By enforcing strict coding standards that ensure optimal performance.
- D. By providing detailed reports on the performance of the codebase.

Answer: A

Explanation:

GitHub Copilot suggests code optimizations by analyzing the codebase and recommending more efficient algorithms or data structures.

Reference: GitHub Copilot code optimization documentation.

Question: 63

Which GitHub Copilot plan allows for prompt and suggestion collection?

- A. GitHub Copilot Individuals
- B. GitHub Copilot Business
- C. GitHub Copilot Enterprise
- D. GitHub Copilot Codespace

Answer: C

Explanation:

GitHub Copilot Enterprise allows for prompt and suggestion collection, enabling organizations to analyze and improve their usage of the tool.

Reference: GitHub Copilot Enterprise data collection documentation.

Question: 64

How does GitHub Copilot Enterprise assist in code reviews during the pull request process? (Select two.)

- A. It automatically merges pull requests after an automated review.
- B. It generates a prose summary and a bulleted list of key changes for pull requests.
- C. It can validate the accuracy of the changes in the pull request.
- D. It can answer questions about the changeset of the pull request.

Answer: B, D

Explanation:

GitHub Copilot Enterprise assists in code reviews by generating summaries of pull requests and answering questions about the changes made.

Reference: GitHub Copilot Enterprise pull request review documentation.

Question: 65

Which of the following statements best describes the impact of GitHub Copilot on the software development process?

- A. It decreases software vulnerabilities from third party dependencies.
- B. It reduces overhead by automating testing workflows.
- C. It increases productivity by automating repetitive coding tasks.
- D. It replaces the need for developers in the software development process.

Answer: C

Explanation:

GitHub Copilot primarily impacts the software development process by increasing productivity through automating repetitive coding tasks.

Reference: GitHub Copilot impact documentation.