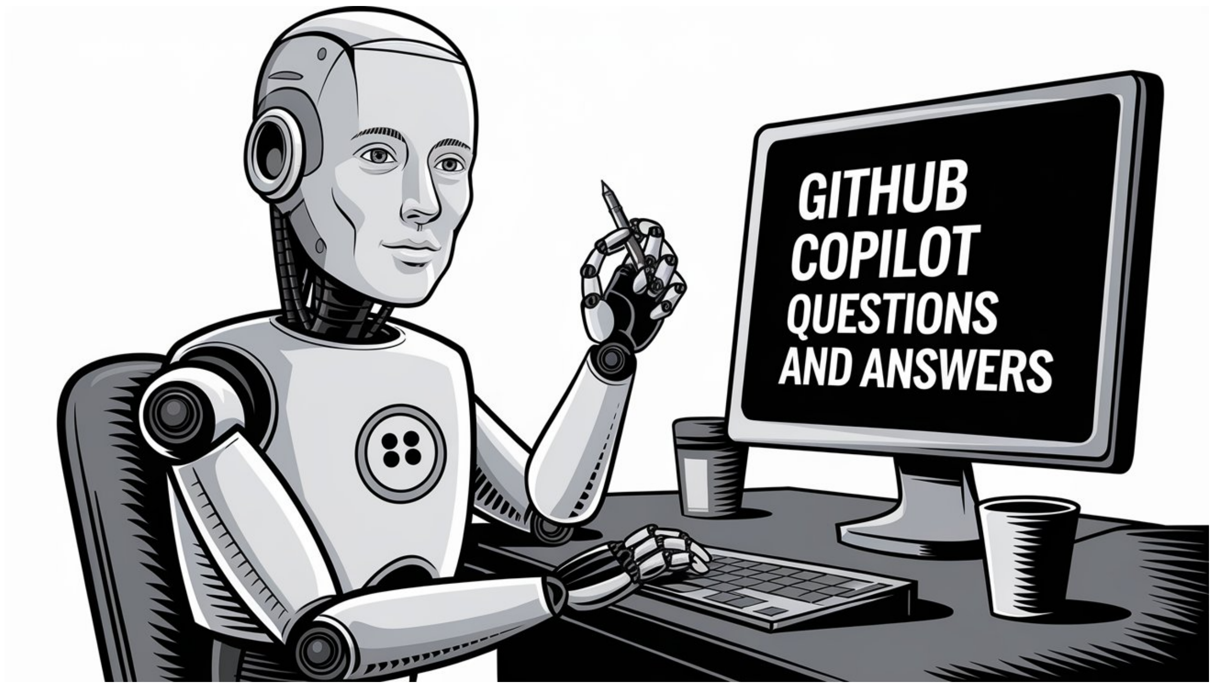
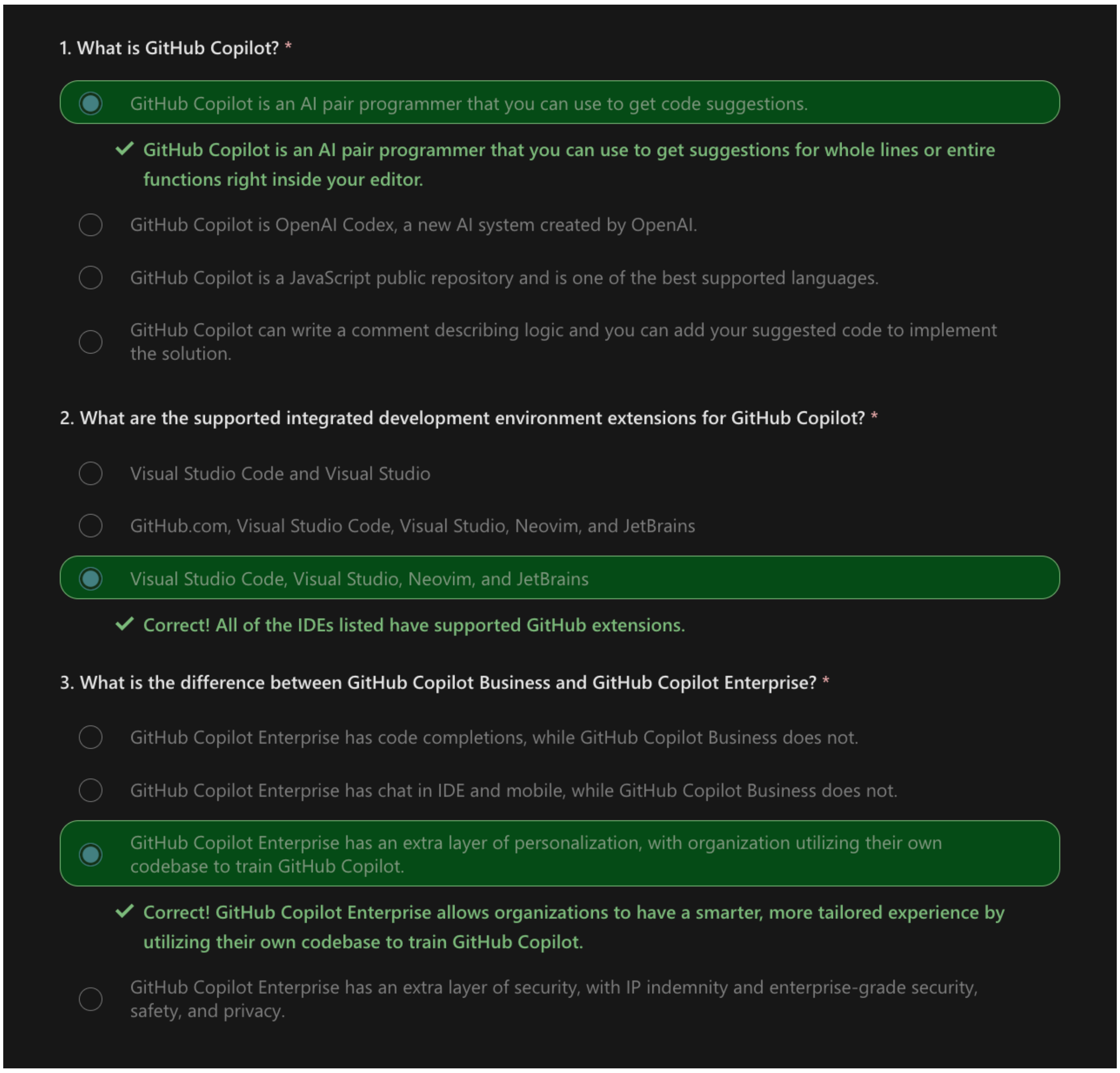


By

Mahendran Selvakumar

https://devopstronaut.com/

https://www.linkedin.com/in/mahendran-selvakumar-36444a77/



1. What is GitHub Copilot? *

[✓ GitHub Copilot is an AI pair programmer that you can use to get code suggestions.]

✓ GitHub Copilot is an AI pair programmer that you can use to get suggestions for whole lines or entire functions right inside your editor.

- GitHub Copilot is OpenAI Codex, a new AI system created by OpenAI.
- GitHub Copilot is a JavaScript public repository and is one of the best supported languages.
- GitHub Copilot can write a comment describing logic and you can add your suggested code to implement the solution.

2. What are the supported integrated development environment extensions for GitHub Copilot? *

- Visual Studio Code and Visual Studio
- GitHub.com, Visual Studio Code, Visual Studio, Neovim, and JetBrains

[✓ Visual Studio Code, Visual Studio, Neovim, and JetBrains]

✓ Correct! All of the IDEs listed have supported GitHub extensions.

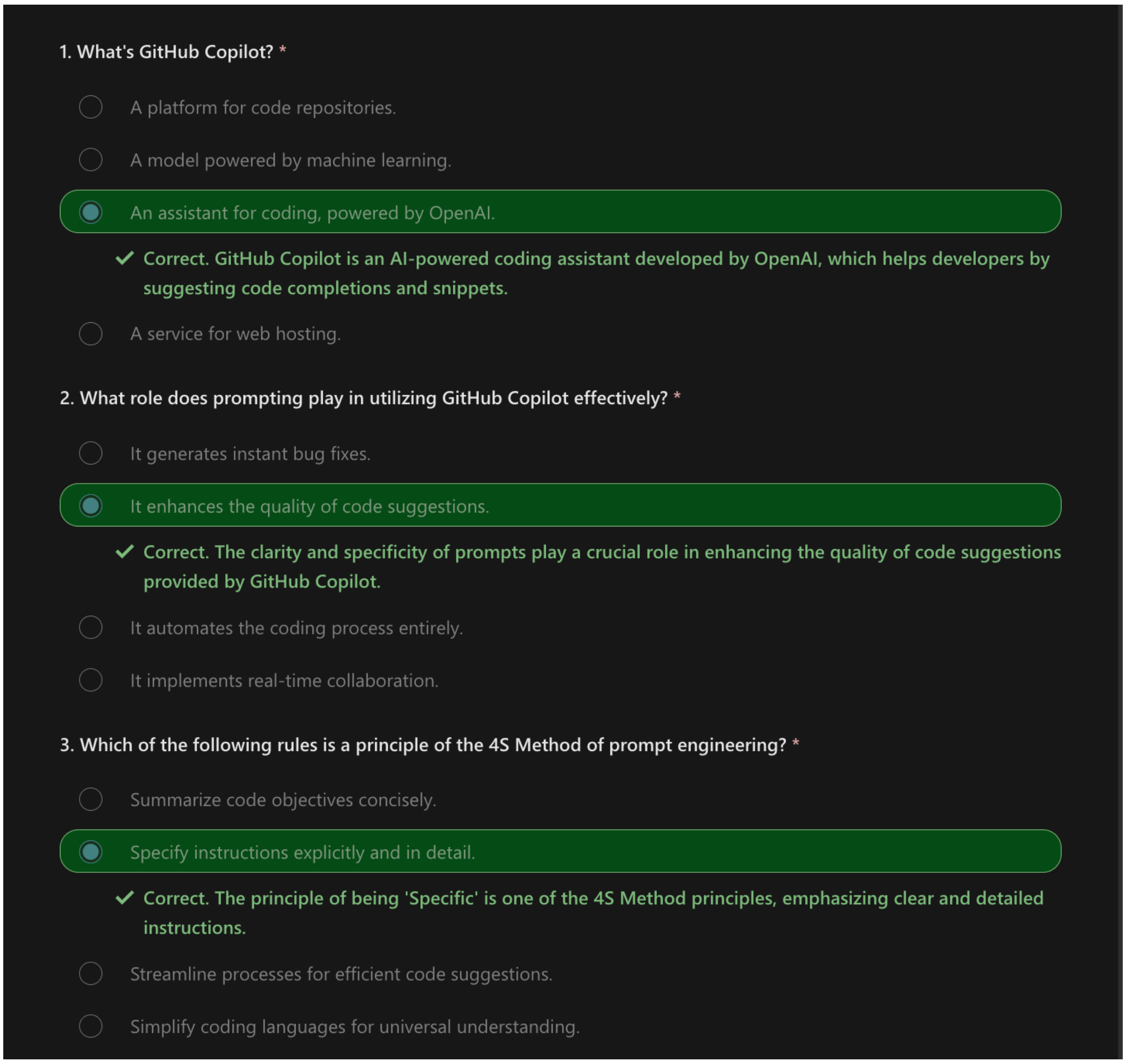
3. What is the difference between GitHub Copilot Business and GitHub Copilot Enterprise? *

- GitHub Copilot Enterprise has code completions, while GitHub Copilot Business does not.
- GitHub Copilot Enterprise has chat in IDE and mobile, while GitHub Copilot Business does not.

[✓ GitHub Copilot Enterprise has an extra layer of personalization, with organization utilizing their own codebase to train GitHub Copilot.]

✓ Correct! GitHub Copilot Enterprise allows organizations to have a smarter, more tailored experience by utilizing their own codebase to train GitHub Copilot.

- GitHub Copilot Enterprise has an extra layer of security, with IP indemnity and enterprise-grade security, safety, and privacy.



1. What's GitHub Copilot? *

- ○ A platform for code repositories.
- ○ A model powered by machine learning.
- ● An assistant for coding, powered by OpenAI.

  ✓ Correct. GitHub Copilot is an AI-powered coding assistant developed by OpenAI, which helps developers by suggesting code completions and snippets.
- ○ A service for web hosting.

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

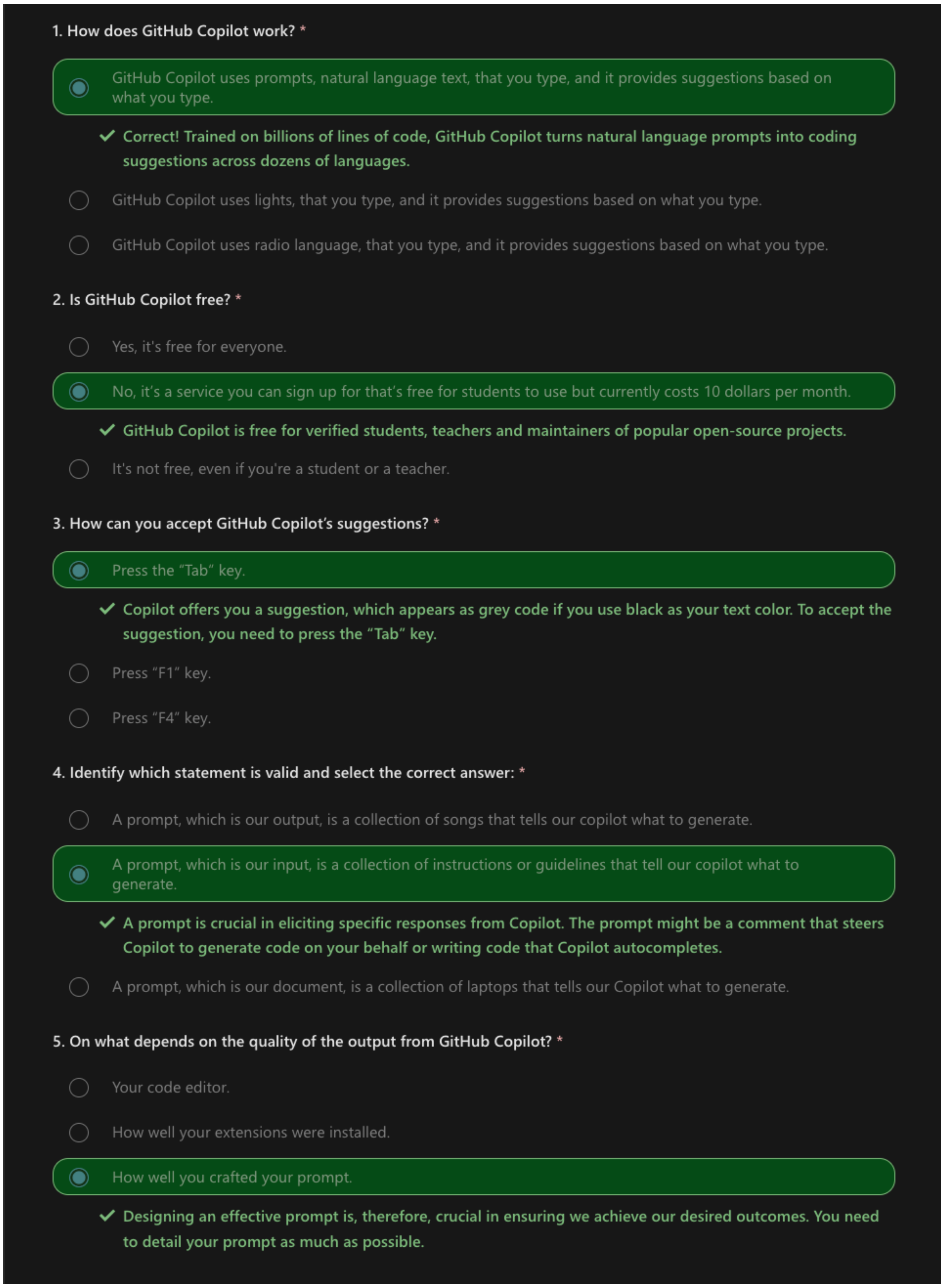
- ○ It generates instant bug fixes.
- ● It enhances the quality of code suggestions.

  ✓ Correct. The clarity and specificity of prompts play a crucial role in enhancing the quality of code suggestions provided by GitHub Copilot.
- ○ It automates the coding process entirely.
- ○ It implements real-time collaboration.

3. Which of the following rules is a principle of the 4S Method of prompt engineering? *

- ○ Summarize code objectives concisely.
- ● Specify instructions explicitly and in detail.

  ✓ Correct. The principle of being 'Specific' is one of the 4S Method principles, emphasizing clear and detailed instructions.
- ○ Streamline processes for efficient code suggestions.
- ○ Simplify coding languages for universal understanding.



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.]

  ✓ Correct! Trained on billions of lines of code, GitHub Copilot turns natural language prompts into coding suggestions across dozens of languages.

- 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.]

  ✓ GitHub Copilot is free for verified students, teachers and maintainers of popular open-source projects.

- 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.]

  ✓ Copilot offers you a suggestion, which appears as grey code if you use black as your text color. To accept the suggestion, you need to press the "Tab" key.

- Press "F1" 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.]

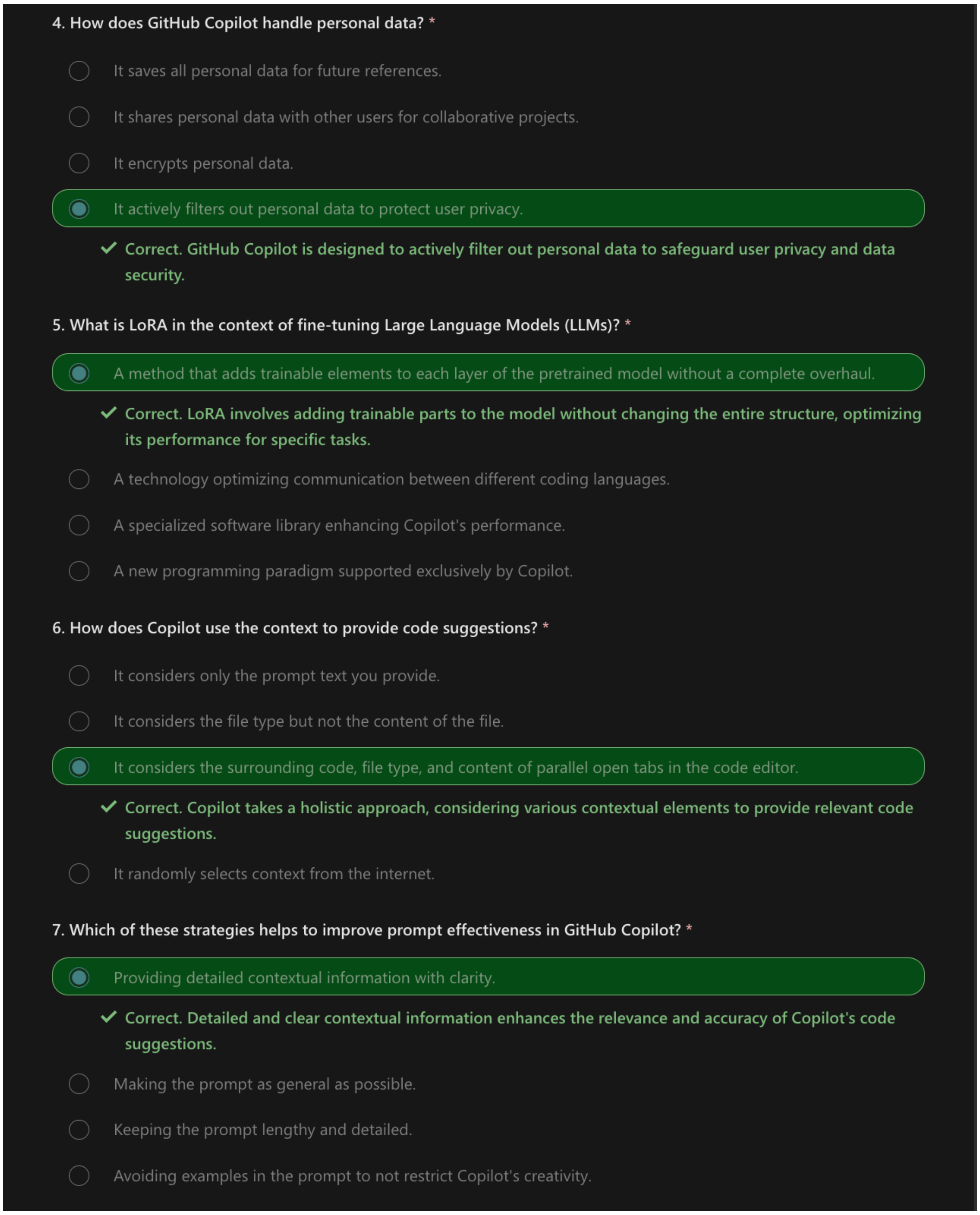
  ✓ A prompt is crucial in eliciting specific responses from Copilot. The prompt might be a comment that steers Copilot to generate code on your behalf or writing code that Copilot autocompletes.

- 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.]

  ✓ Designing an effective prompt is, therefore, crucial in ensuring we achieve our desired outcomes. You need to detail your prompt as much as possible.



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.]

✓ Correct. GitHub Copilot is designed to actively filter out personal data to safeguard user privacy and data security.

5. What is LoRA in the context of fine-tuning Large Language Models (LLMs)? *

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

✓ Correct. LoRA involves adding trainable parts to the model without changing the entire structure, optimizing its performance for specific tasks.

- 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.]

✓ Correct. Copilot takes a holistic approach, considering various contextual elements to provide relevant code suggestions.

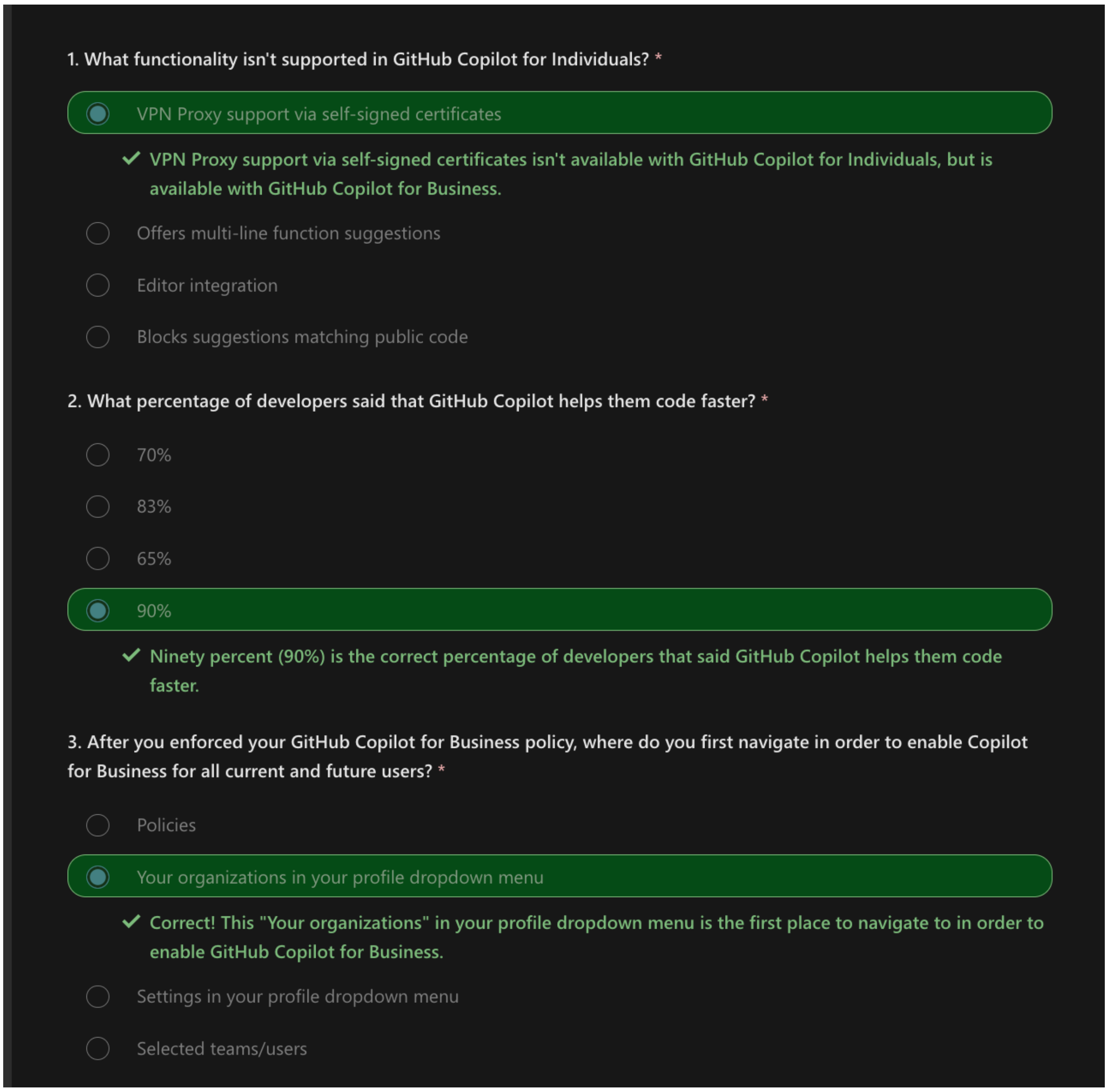
- It randomly selects context from the internet.

7. Which of these strategies helps to improve prompt effectiveness in GitHub Copilot? *

- [✓ Providing detailed contextual information with clarity.]

✓ Correct. Detailed and clear contextual information enhances the relevance and accuracy of Copilot's code suggestions.

- Making the prompt as general as possible.
- Keeping the prompt lengthy and detailed.
- Avoiding examples in the prompt to not restrict Copilot's creativity.



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

[✓ VPN Proxy support via self-signed certificates]

✓ VPN Proxy support via self-signed certificates isn't available with GitHub Copilot for Individuals, but is available with GitHub Copilot for Business.

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%]

✓ Ninety percent (90%) is the correct percentage of developers that said GitHub Copilot helps them code faster.

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

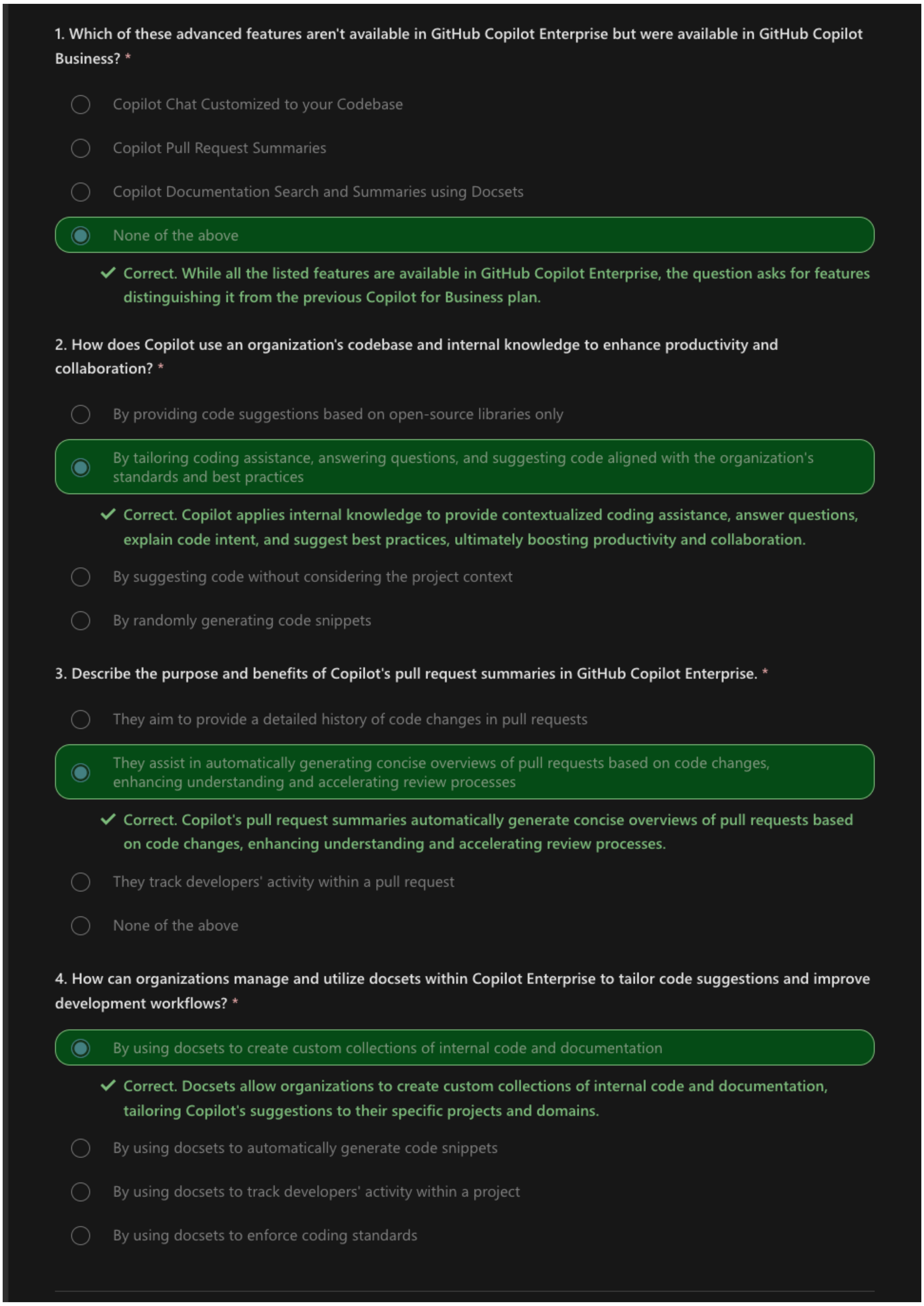
Policies

[✓ Your organizations in your profile dropdown menu]

✓ Correct! This "Your organizations" in your profile dropdown menu is the first place to navigate to in order to enable GitHub Copilot for Business.

Settings in your profile dropdown menu

Selected teams/users



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]

✓ Correct. While all the listed features are available in GitHub Copilot Enterprise, the question asks for features distinguishing it from the previous Copilot for Business plan.

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]

✓ Correct. Copilot applies internal knowledge to provide contextualized coding assistance, answer questions, explain code intent, and suggest best practices, ultimately boosting productivity and collaboration.

- ○ 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]

✓ Correct. Copilot's pull request summaries automatically generate concise overviews of pull requests based on code changes, enhancing understanding and accelerating review processes.

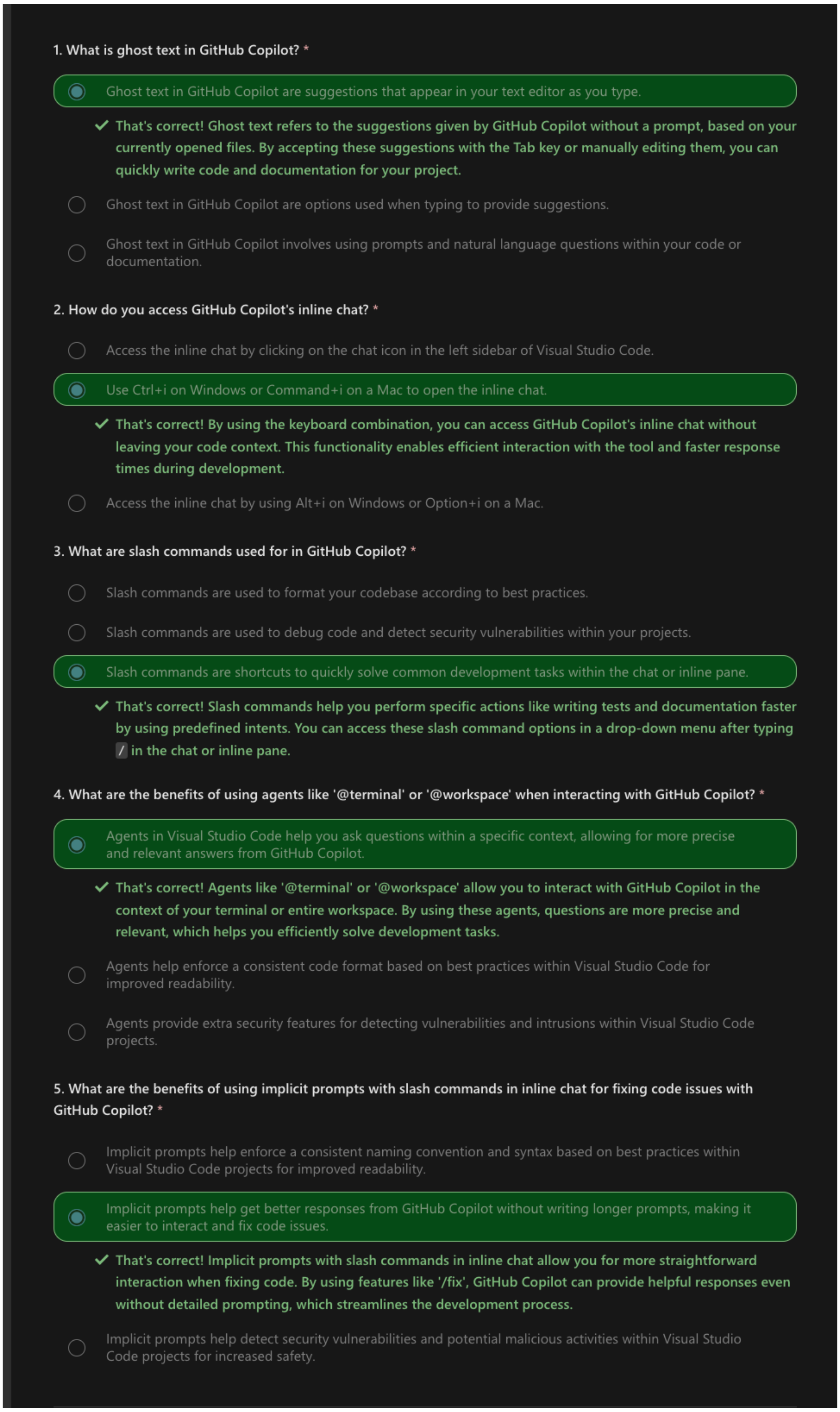
- ○ They track developers' activity within a pull request
- ○ None of the above

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]

✓ Correct. Docsets allow organizations to create custom collections of internal code and documentation, tailoring Copilot's suggestions to their specific projects and domains.

- ○ 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



1. What is ghost text in GitHub Copilot? *

[✓ Ghost text in GitHub Copilot are suggestions that appear in your text editor as you type.]

✓ That's correct! Ghost text refers to the suggestions given by GitHub Copilot without a prompt, based on your currently opened files. By accepting these suggestions with the Tab key or manually editing them, you can quickly write code and documentation for your project.

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.]

✓ That's correct! By using the keyboard combination, you can access GitHub Copilot's inline chat without leaving your code context. This functionality enables efficient interaction with the tool and faster response times during development.

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.]

✓ That's correct! Slash commands help you perform specific actions like writing tests and documentation faster by using predefined intents. You can access these slash command options in a drop-down menu after typing `/` in 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.]

✓ That's correct! Agents like '@terminal' or '@workspace' allow you to interact with GitHub Copilot in the context of your terminal or entire workspace. By using these agents, questions are more precise and relevant, which helps you efficiently solve development tasks.

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.

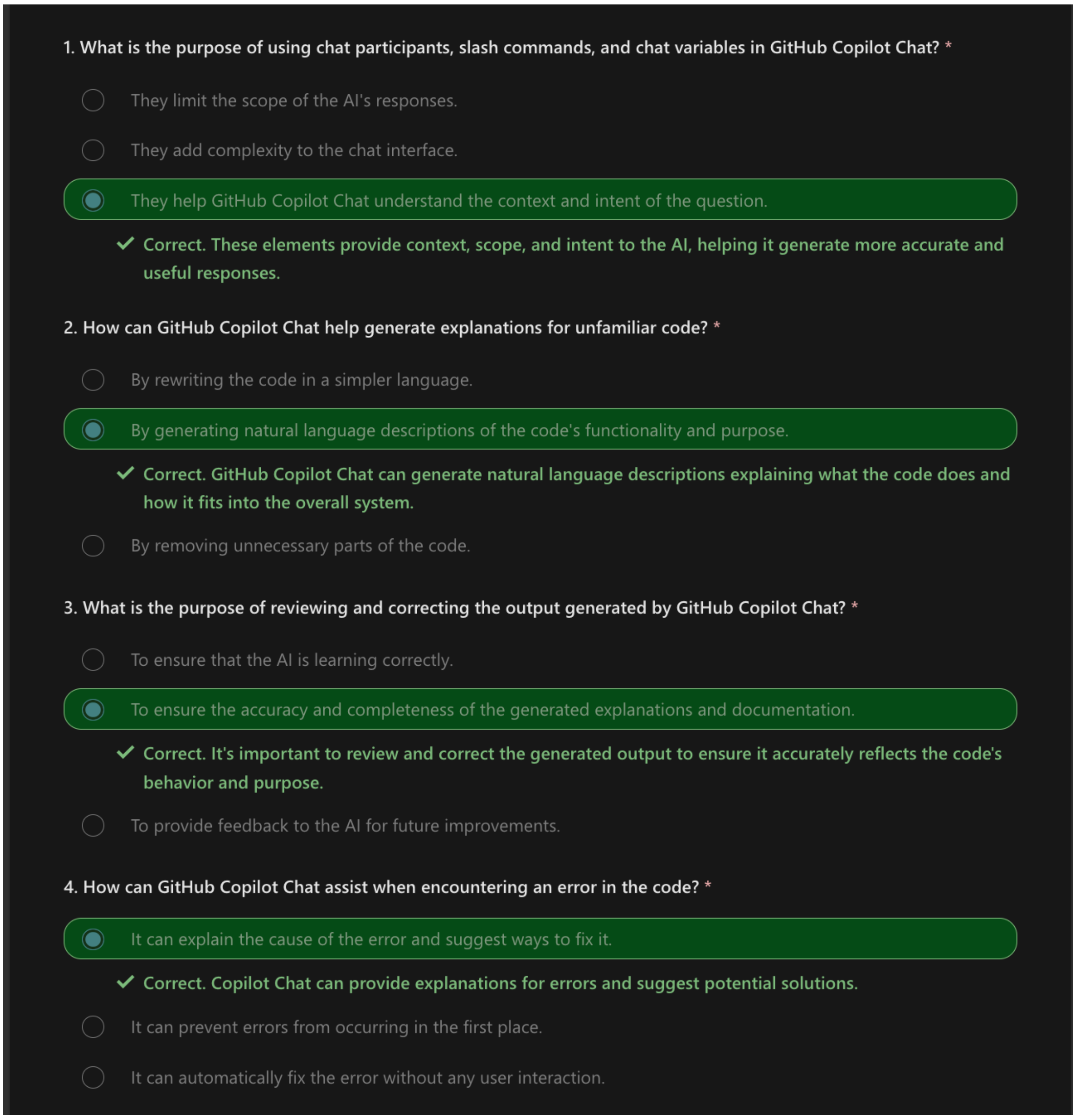
5. What are the benefits of using implicit prompts with slash commands in inline chat for fixing code issues with GitHub Copilot? *

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.]

✓ That's correct! Implicit prompts with slash commands in inline chat allow you for more straightforward interaction when fixing code. By using features like '/fix', GitHub Copilot can provide helpful responses even without detailed prompting, which streamlines the development process.

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


1. What is the purpose of using chat participants, slash commands, and chat variables in GitHub Copilot Chat? *

- ○ They limit the scope of the AI's responses.
- ○ They add complexity to the chat interface.
- [✓ They help GitHub Copilot Chat understand the context and intent of the question.]

✓ Correct. These elements provide context, scope, and intent to the AI, helping it generate more accurate and useful responses.

2. How can GitHub Copilot Chat help generate explanations for unfamiliar code? *

- ○ By rewriting the code in a simpler language.
- [✓ By generating natural language descriptions of the code's functionality and purpose.]

✓ Correct. GitHub Copilot Chat can generate natural language descriptions explaining what the code does and how it fits into the overall system.

- ○ By removing unnecessary parts of the code.

3. What is the purpose of reviewing and correcting the output generated by GitHub Copilot Chat? *

- ○ To ensure that the AI is learning correctly.
- [✓ To ensure the accuracy and completeness of the generated explanations and documentation.]

✓ Correct. It's important to review and correct the generated output to ensure it accurately reflects the code's behavior and purpose.

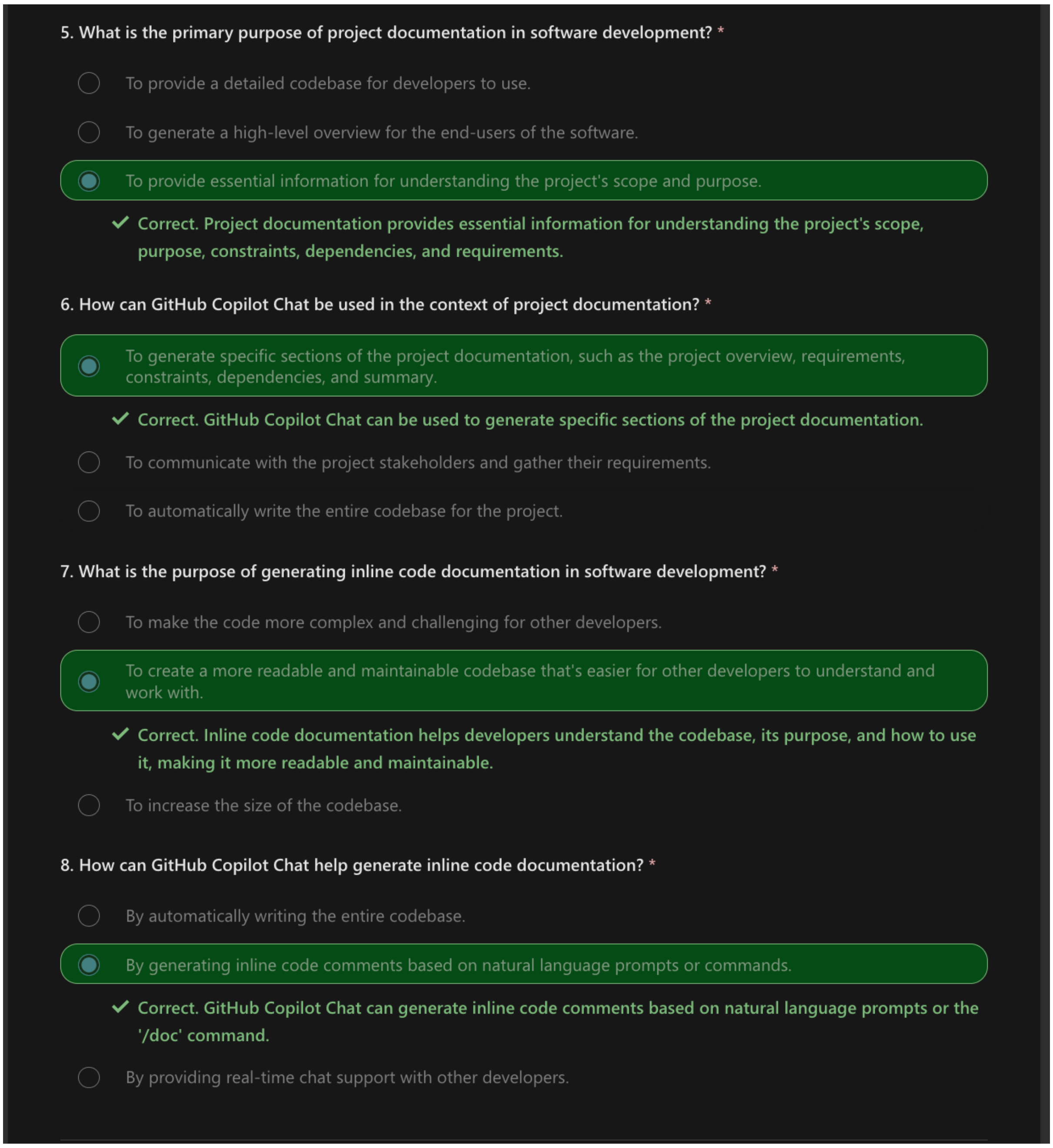
- ○ To provide feedback to the AI for future improvements.

4. How can GitHub Copilot Chat assist when encountering an error in the code? *

- [✓ It can explain the cause of the error and suggest ways to fix it.]

✓ Correct. Copilot Chat can provide explanations for errors and suggest potential solutions.

- ○ It can prevent errors from occurring in the first place.
- ○ It can automatically fix the error without any user interaction.



5. What is the primary purpose of project documentation in software development? *

- ○ To provide a detailed codebase for developers to use.
- ○ To generate a high-level overview for the end-users of the software.
- [✓ To provide essential information for understanding the project's scope and purpose.]

✓ Correct. Project documentation provides essential information for understanding the project's scope, purpose, constraints, dependencies, and requirements.

6. How can GitHub Copilot Chat be used in the context of project documentation? *

- [✓ To generate specific sections of the project documentation, such as the project overview, requirements, constraints, dependencies, and summary.]

✓ Correct. GitHub Copilot Chat can be used to generate specific sections of the project documentation.

- ○ To communicate with the project stakeholders and gather their requirements.
- ○ To automatically write the entire codebase for the project.

7. What is the purpose of generating inline code documentation in software development? *

- ○ To make the code more complex and challenging for other developers.
- [✓ To create a more readable and maintainable codebase that's easier for other developers to understand and work with.]

✓ Correct. Inline code documentation helps developers understand the codebase, its purpose, and how to use it, making it more readable and maintainable.

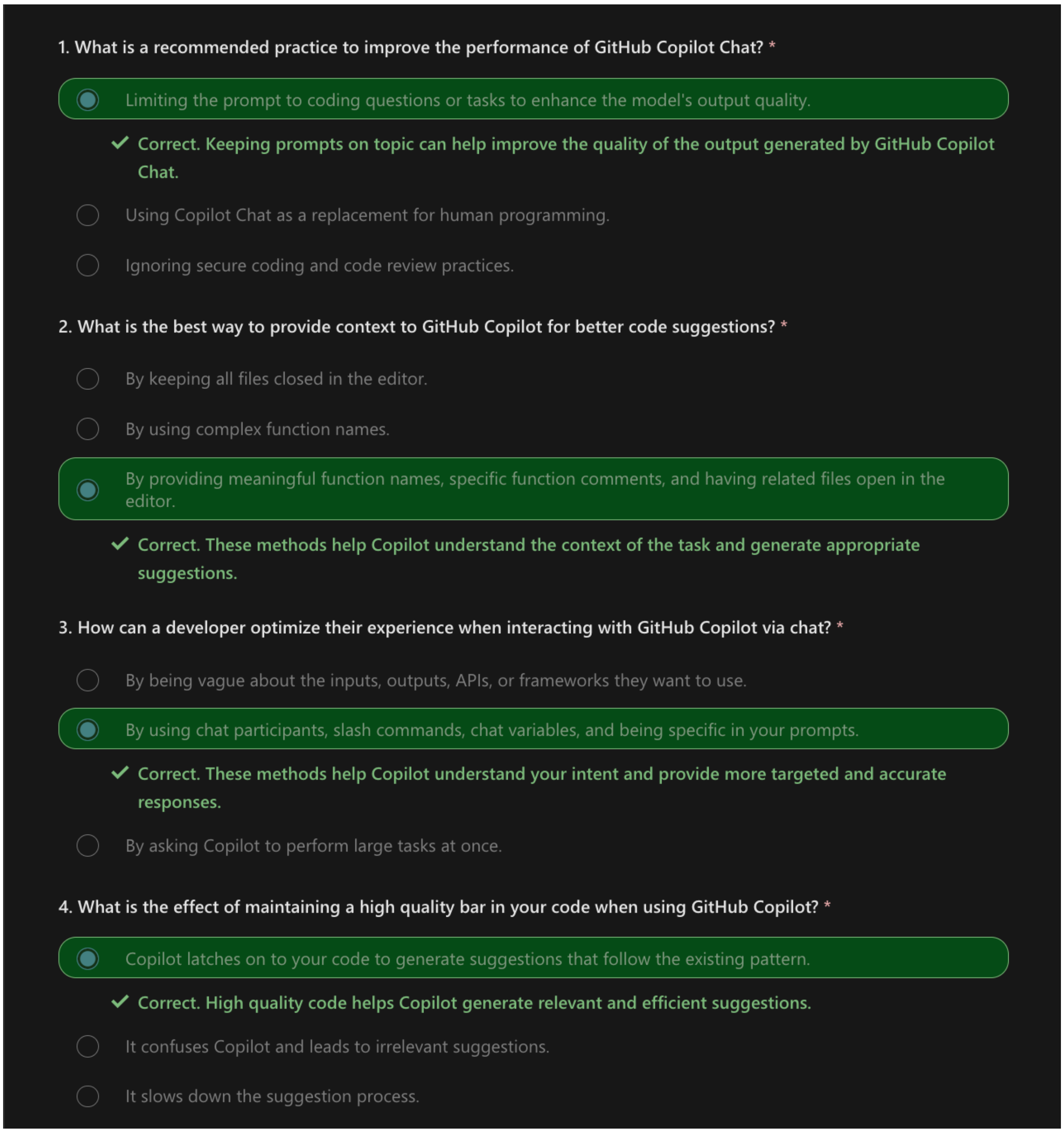
- ○ To increase the size of the codebase.

8. How can GitHub Copilot Chat help generate inline code documentation? *

- ○ By automatically writing the entire codebase.
- [✓ By generating inline code comments based on natural language prompts or commands.]

✓ Correct. GitHub Copilot Chat can generate inline code comments based on natural language prompts or the '/doc' command.

- ○ By providing real-time chat support with other developers.



1. What is a recommended practice to improve the performance of GitHub Copilot Chat? *

[✓ Limiting the prompt to coding questions or tasks to enhance the model's output quality.]

✓ Correct. Keeping prompts on topic can help improve the quality of the output generated by GitHub Copilot Chat.

Using Copilot Chat as a replacement for human programming.

Ignoring secure coding and code review practices.

2. What is the best way to provide context to GitHub Copilot for better code suggestions? *

By keeping all files closed in the editor.

By using complex function names.

[✓ By providing meaningful function names, specific function comments, and having related files open in the editor.]

✓ Correct. These methods help Copilot understand the context of the task and generate appropriate suggestions.

3. How can a developer optimize their experience when interacting with GitHub Copilot via chat? *

By being vague about the inputs, outputs, APIs, or frameworks they want to use.

[✓ By using chat participants, slash commands, chat variables, and being specific in your prompts.]

✓ Correct. These methods help Copilot understand your intent and provide more targeted and accurate responses.

By asking Copilot to perform large tasks at once.

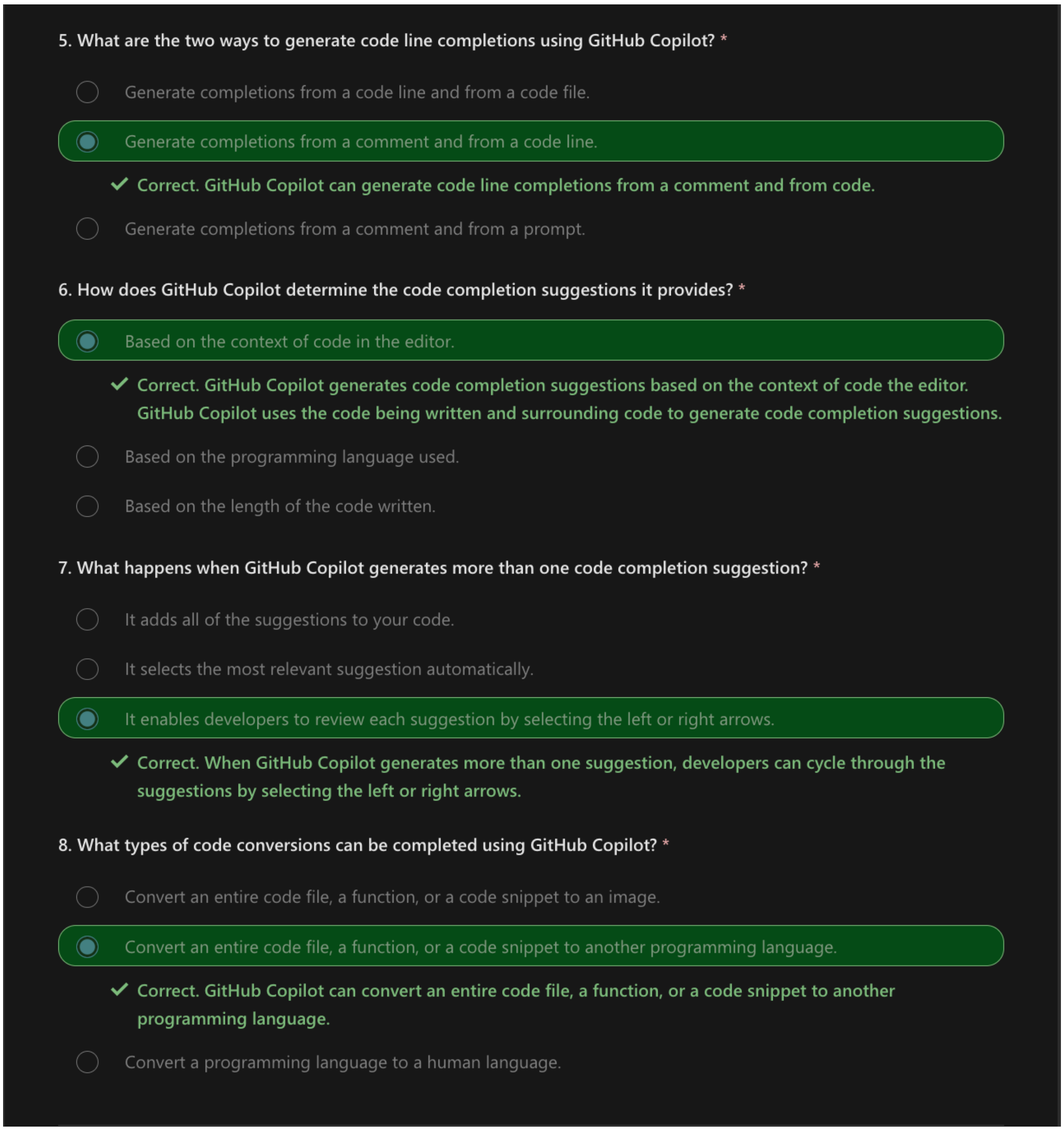
4. What is the effect of maintaining a high quality bar in your code when using GitHub Copilot? *

[✓ Copilot latches on to your code to generate suggestions that follow the existing pattern.]

✓ Correct. High quality code helps Copilot generate relevant and efficient suggestions.

It confuses Copilot and leads to irrelevant suggestions.

It slows down the suggestion process.



5. What are the two ways to generate code line completions using GitHub Copilot? *

- ○ Generate completions from a code line and from a code file.
- [● Generate completions from a comment and from a code line.]

✓ Correct. GitHub Copilot can generate code line completions from a comment and from code.

- ○ Generate completions from a comment and from a prompt.

6. How does GitHub Copilot determine the code completion suggestions it provides? *

- [● Based on the context of code in the editor.]

✓ Correct. GitHub Copilot generates code completion suggestions based on the context of code the editor. GitHub Copilot uses the code being written and surrounding code to generate code completion suggestions.

- ○ Based on the programming language used.
- ○ Based on the length of the code written.

7. What happens when GitHub Copilot generates more than one code completion suggestion? *

- ○ It adds all of the suggestions to your code.
- ○ It selects the most relevant suggestion automatically.
- [● It enables developers to review each suggestion by selecting the left or right arrows.]

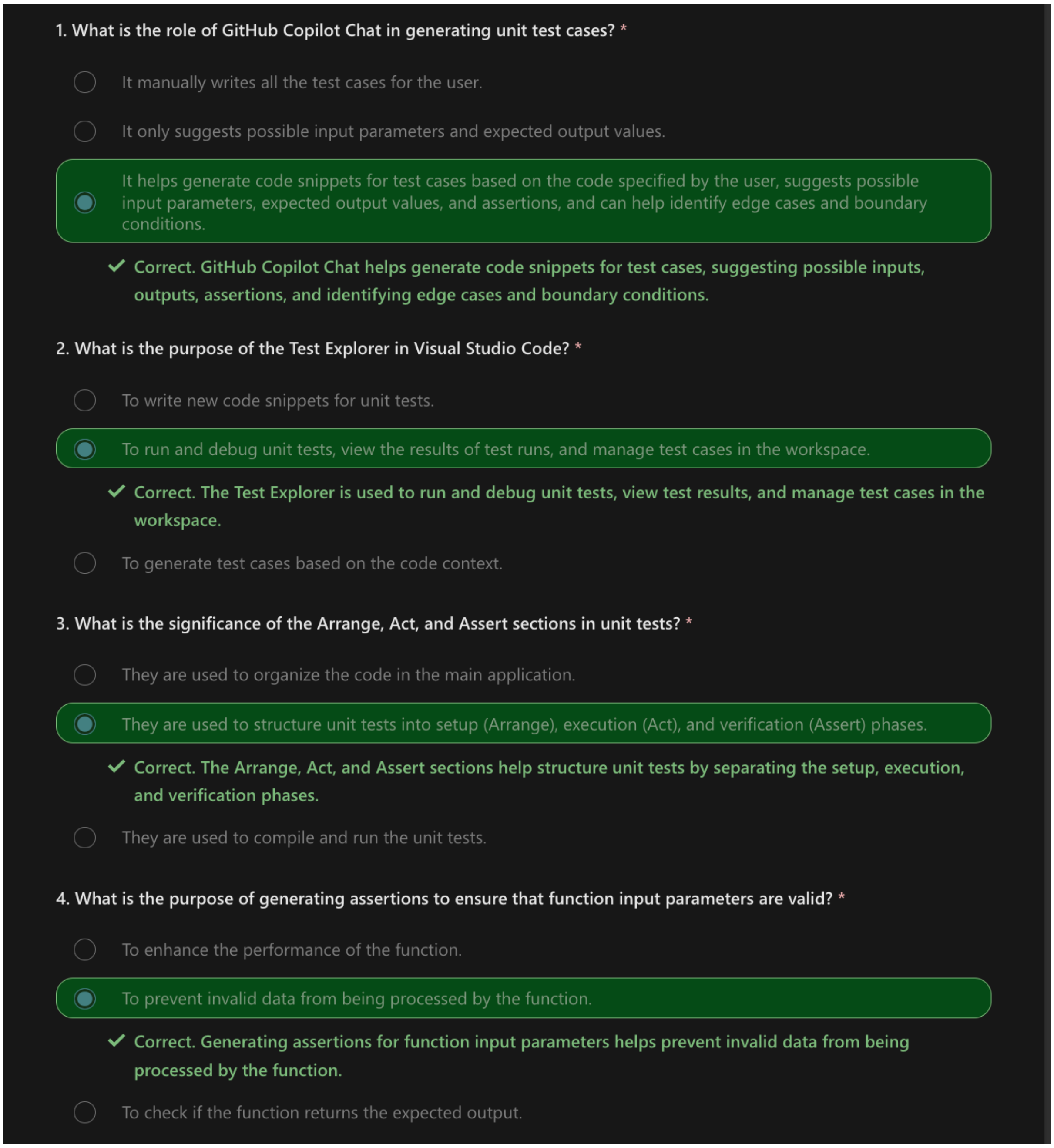
✓ Correct. When GitHub Copilot generates more than one suggestion, developers can cycle through the suggestions by selecting the left or right arrows.

8. What types of code conversions can be completed using GitHub Copilot? *

- ○ Convert an entire code file, a function, or a code snippet to an image.
- [● Convert an entire code file, a function, or a code snippet to another programming language.]

✓ Correct. GitHub Copilot can convert an entire code file, a function, or a code snippet to another programming language.

- ○ Convert a programming language to a human language.



1. What is the role of GitHub Copilot Chat in generating unit test cases? *

- ○ It manually writes all the test cases for the user.
- ○ It only suggests possible input parameters and expected output values.
- [✓ It helps generate code snippets for test cases based on the code specified by the user, suggests possible input parameters, expected output values, and assertions, and can help identify edge cases and boundary conditions.]

✓ Correct. GitHub Copilot Chat helps generate code snippets for test cases, suggesting possible inputs, outputs, assertions, and identifying edge cases and boundary conditions.

2. What is the purpose of the Test Explorer in Visual Studio Code? *

- ○ To write new code snippets for unit tests.
- [✓ To run and debug unit tests, view the results of test runs, and manage test cases in the workspace.]

✓ Correct. The Test Explorer is used to run and debug unit tests, view test results, and manage test cases in the workspace.

- ○ To generate test cases based on the code context.

3. What is the significance of the Arrange, Act, and Assert sections in unit tests? *

- ○ They are used to organize the code in the main application.
- [✓ They are used to structure unit tests into setup (Arrange), execution (Act), and verification (Assert) phases.]

✓ Correct. The Arrange, Act, and Assert sections help structure unit tests by separating the setup, execution, and verification phases.

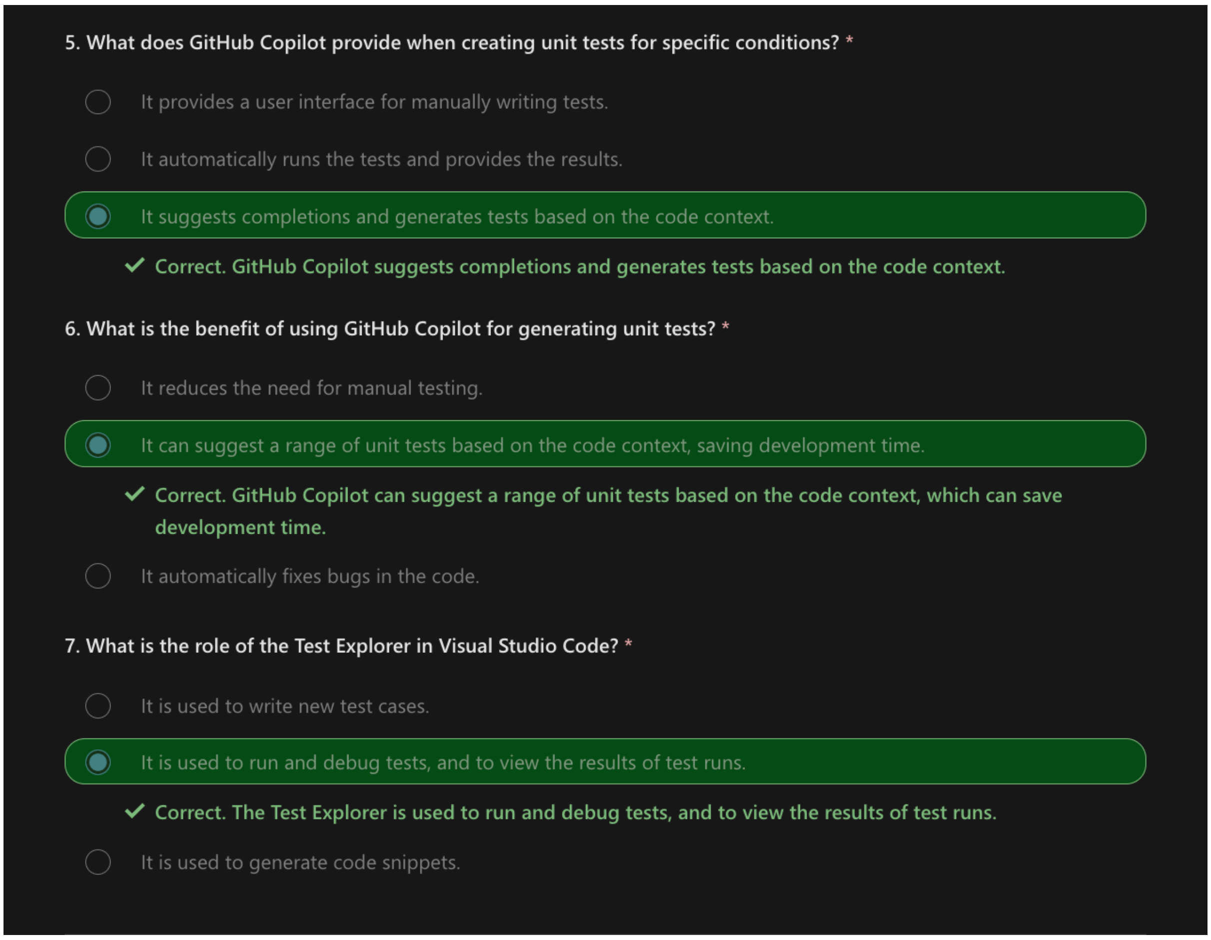
- ○ They are used to compile and run the unit tests.

4. What is the purpose of generating assertions to ensure that function input parameters are valid? *

- ○ To enhance the performance of the function.
- [✓ To prevent invalid data from being processed by the function.]

✓ Correct. Generating assertions for function input parameters helps prevent invalid data from being processed by the function.

- ○ To check if the function returns the expected output.



5. What does GitHub Copilot provide when creating unit tests for specific conditions? *

- ○ It provides a user interface for manually writing tests.
- ○ It automatically runs the tests and provides the results.
- [✓ It suggests completions and generates tests based on the code context.]

✓ Correct. GitHub Copilot suggests completions and generates tests based on the code context.

6. What is the benefit of using GitHub Copilot for generating unit tests? *

- ○ It reduces the need for manual testing.
- [✓ It can suggest a range of unit tests based on the code context, saving development time.]

✓ Correct. GitHub Copilot can suggest a range of unit tests based on the code context, which can save development time.

- ○ It automatically fixes bugs in the code.

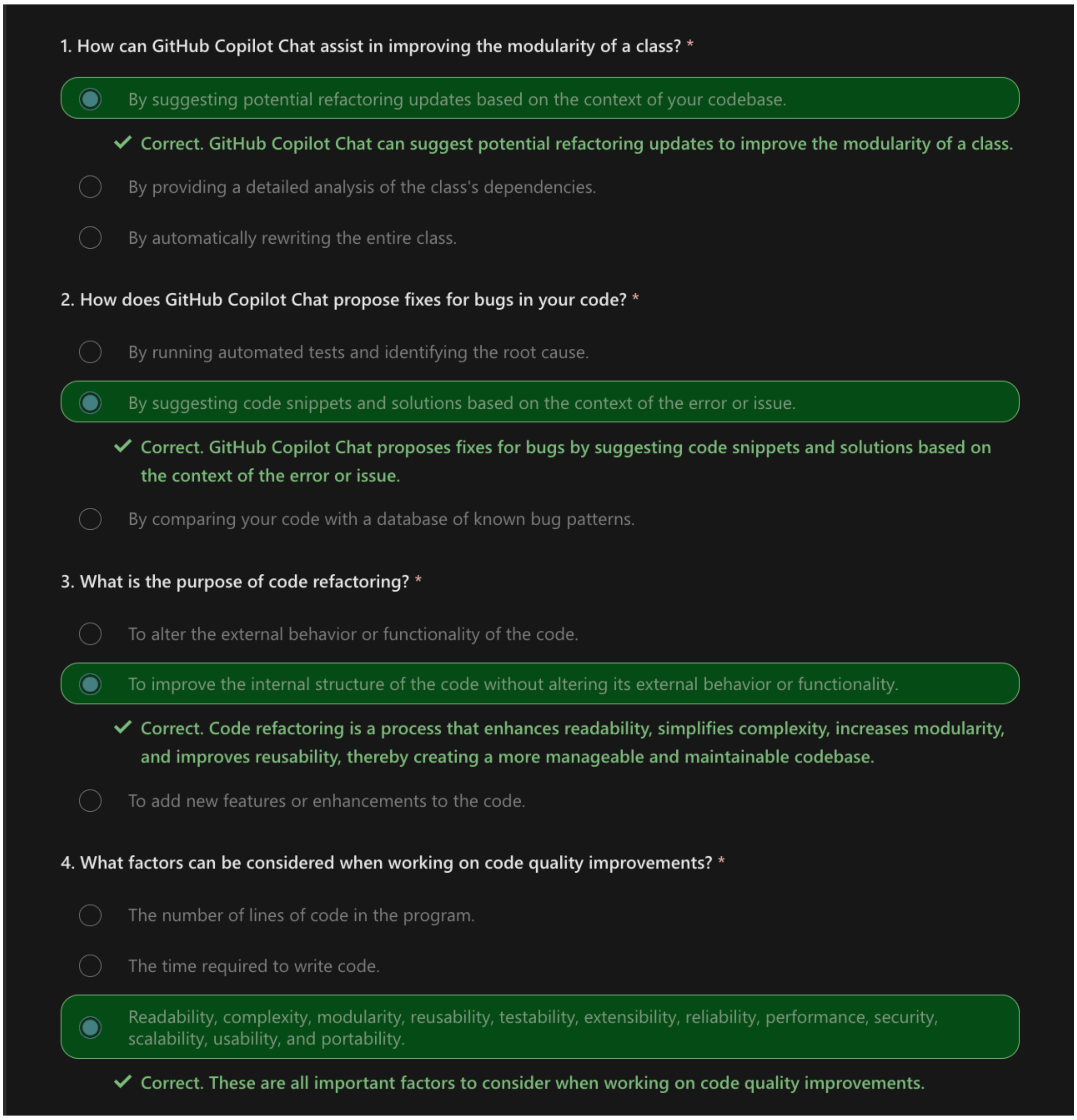
7. What is the role of the Test Explorer in Visual Studio Code? *

- ○ It is used to write new test cases.
- [✓ It is used to run and debug tests, and to view the results of test runs.]

✓ Correct. The Test Explorer is used to run and debug tests, and to view the results of test runs.

- ○ It is used to generate code snippets.

https://www.linkedin.com/in/mahendran-selvakumar-36444a77/



1. How can GitHub Copilot Chat assist in improving the modularity of a class? *

[✓ By suggesting potential refactoring updates based on the context of your codebase.]

✓ Correct. GitHub Copilot Chat can suggest potential refactoring updates to improve the modularity of a class.

By providing a detailed analysis of the class's dependencies.

By automatically rewriting the entire class.

2. How does GitHub Copilot Chat propose fixes for bugs in your code? *

By running automated tests and identifying the root cause.

[✓ By suggesting code snippets and solutions based on the context of the error or issue.]

✓ Correct. GitHub Copilot Chat proposes fixes for bugs by suggesting code snippets and solutions based on the context of the error or issue.

By comparing your code with a database of known bug patterns.

3. What is the purpose of code refactoring? *

To alter the external behavior or functionality of the code.

[✓ To improve the internal structure of the code without altering its external behavior or functionality.]

✓ Correct. Code refactoring is a process that enhances readability, simplifies complexity, increases modularity, and improves reusability, thereby creating a more manageable and maintainable codebase.

To add new features or enhancements to the code.

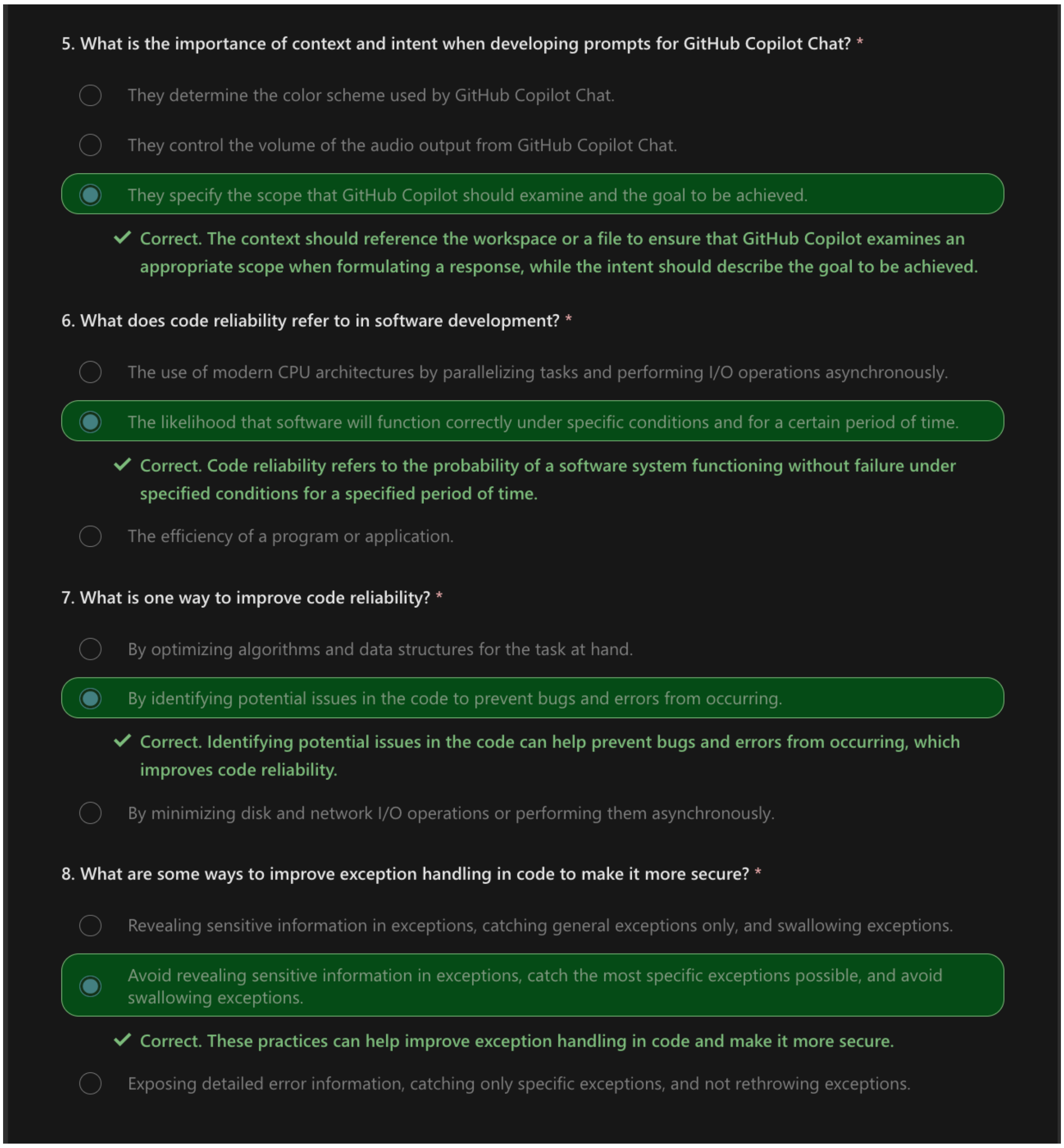
4. What factors can be considered when working on code quality improvements? *

The number of lines of code in the program.

The time required to write code.

[✓ Readability, complexity, modularity, reusability, testability, extensibility, reliability, performance, security, scalability, usability, and portability.]

✓ Correct. These are all important factors to consider when working on code quality improvements.



5. What is the importance of context and intent when developing prompts for GitHub Copilot Chat? *

- They determine the color scheme used by GitHub Copilot Chat.
- They control the volume of the audio output from GitHub Copilot Chat.
- [✓ They specify the scope that GitHub Copilot should examine and the goal to be achieved.]

✓ Correct. The context should reference the workspace or a file to ensure that GitHub Copilot examines an appropriate scope when formulating a response, while the intent should describe the goal to be achieved.

6. What does code reliability refer to in software development? *

- The use of modern CPU architectures by parallelizing tasks and performing I/O operations asynchronously.
- [✓ The likelihood that software will function correctly under specific conditions and for a certain period of time.]

✓ Correct. Code reliability refers to the probability of a software system functioning without failure under specified conditions for a specified period of time.

- The efficiency of a program or application.

7. What is one way to improve code reliability? *

- By optimizing algorithms and data structures for the task at hand.
- [✓ By identifying potential issues in the code to prevent bugs and errors from occurring.]

✓ Correct. Identifying potential issues in the code can help prevent bugs and errors from occurring, which improves code reliability.

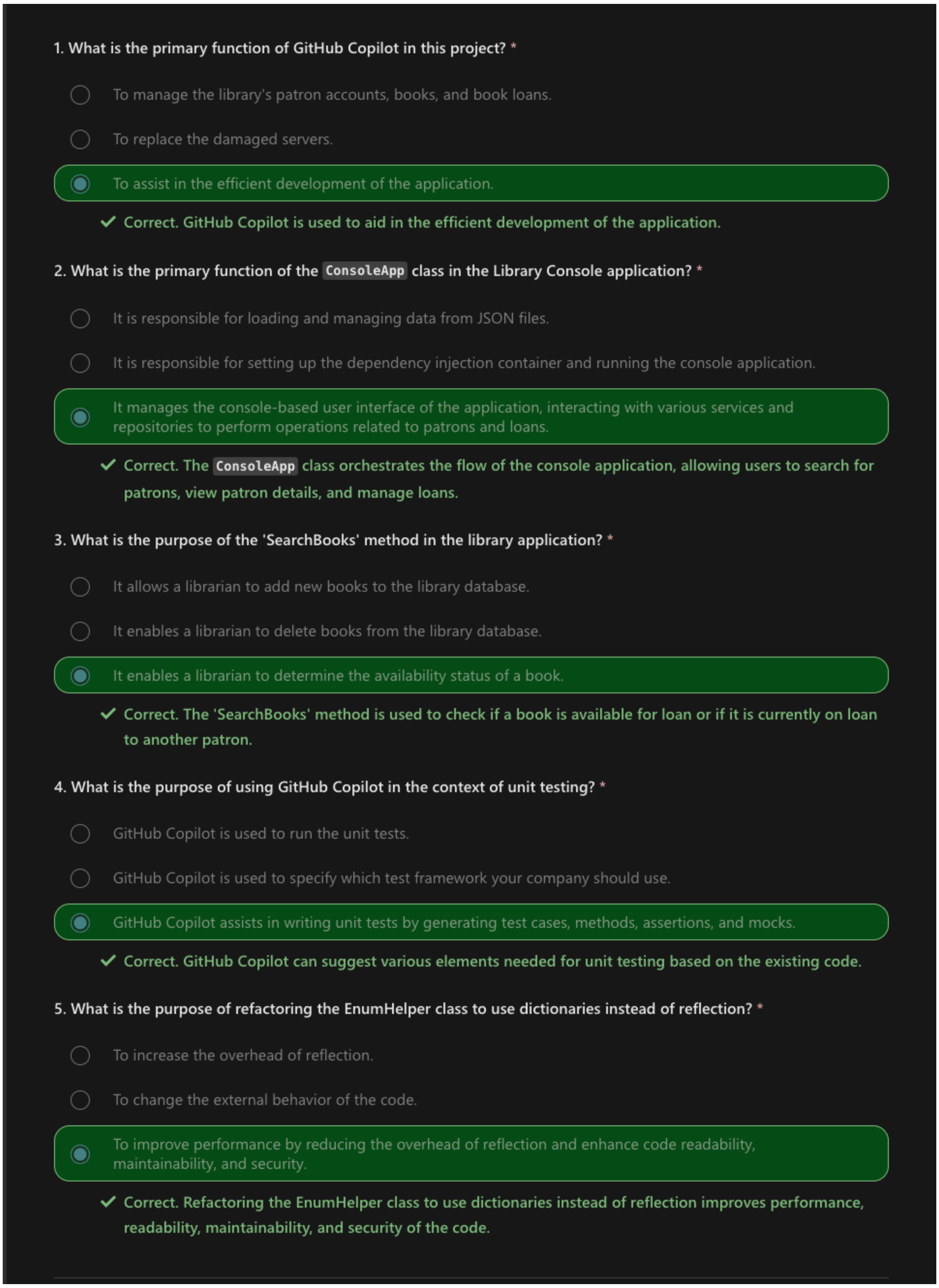
- By minimizing disk and network I/O operations or performing them asynchronously.

8. What are some ways to improve exception handling in code to make it more secure? *

- Revealing sensitive information in exceptions, catching general exceptions only, and swallowing exceptions.
- [✓ Avoid revealing sensitive information in exceptions, catch the most specific exceptions possible, and avoid swallowing exceptions.]

✓ Correct. These practices can help improve exception handling in code and make it more secure.

- Exposing detailed error information, catching only specific exceptions, and not rethrowing exceptions.



1. What is the primary function of GitHub Copilot in this project? *

- To manage the library's patron accounts, books, and book loans.
- To replace the damaged servers.
- [✓ To assist in the efficient development of the application.]

✓ Correct. GitHub Copilot is used to aid in the efficient development of the application.

2. What is the primary function of the `ConsoleApp` class in the Library Console application? *

- It is responsible for loading and managing data from JSON files.
- It is responsible for setting up the dependency injection container and running the console application.
- [✓ It manages the console-based user interface of the application, interacting with various services and repositories to perform operations related to patrons and loans.]

✓ Correct. The `ConsoleApp` class orchestrates the flow of the console application, allowing users to search for patrons, view patron details, and manage loans.

3. What is the purpose of the 'SearchBooks' method in the library application? *

- It allows a librarian to add new books to the library database.
- It enables a librarian to delete books from the library database.
- [✓ It enables a librarian to determine the availability status of a book.]

✓ Correct. The 'SearchBooks' method is used to check if a book is available for loan or if it is currently on loan to another patron.

4. What is the purpose of using GitHub Copilot in the context of unit testing? *

- GitHub Copilot is used to run the unit tests.
- GitHub Copilot is used to specify which test framework your company should use.
- [✓ GitHub Copilot assists in writing unit tests by generating test cases, methods, assertions, and mocks.]

✓ Correct. GitHub Copilot can suggest various elements needed for unit testing based on the existing code.

5. What is the purpose of refactoring the EnumHelper class to use dictionaries instead of reflection? *

- To increase the overhead of reflection.
- To change the external behavior of the code.
- [✓ To improve performance by reducing the overhead of reflection and enhance code readability, maintainability, and security.]

✓ Correct. Refactoring the EnumHelper class to use dictionaries instead of reflection improves performance, readability, maintainability, and security of the code.