EDUSUM
#1 Online Certification Guide

MICROSOFT GH-300

Microsoft GitHub Copilot Certification Questions & Answers

Exam Summary – Syllabus –Questions

GH-300
Microsoft GitHub Copilot
65 Questions Exam – 700 / 1000 Cut Score – Duration of 100 minutes

Table of Contents:

Know Your GH-300 Certification Well:

The GH-300 is best suitable for candidates who want to gain knowledge in the Microsoft GitHub. Before you start your GH-300 preparation you may struggle to get all the crucial GitHub Copilot materials like GH-300 syllabus, sample questions, study guide.

But don't worry the GH-300 PDF is here to help you prepare in a stress free manner.
The PDF is a combination of all your queries like-

- What is in the GH-300 syllabus?
- How many questions are there in the GH-300 exam?
- Which Practice test would help me to pass the GH-300 exam at the first attempt?

Passing the GH-300 exam makes you Microsoft GitHub Copilot. Having the GitHub Copilot certification opens multiple opportunities for you. You can grab a new job, get a higher salary or simply get recognition within your current organization.

Microsoft GH-300 GitHub Copilot Certification Details:

Exam Name	Microsoft GitHub Copilot
Exam Code	GH-300
Exam Price	$99 (USD)
Duration	100 mins
Number of Questions	65
Passing Score	700 / 1000
Books / Training	**GH-300T00-A: GitHub Copilot**
Schedule Exam	**Pearson VUE**
Sample Questions	**Microsoft GitHub Copilot Sample Questions**
Practice Exam	**Microsoft GH-300 Certification Practice Exam**

GH-300 Syllabus:

Topic	**Details**
Responsible AI (7%)	
Explain responsible usage of AI	- Describe the risks associated with using AI - Explain the limitations of using generative AI tools (depth of the source data for the model, bias in the data, etc.) - Explain the need to validate the output of AI tools - Identify how to operate a responsible AI - Identify the potential harms of generative AI (bias, secure code, fairness, privacy, transparency) - Explain how to mitigate the occurrence of potential harms - Explain ethical AI
GitHub Copilot plans and features (31%)	
Identify the different GitHub Copilot plans	- Understand the differences between Copilot Individual, Copilot Business, Copilot Enterprise, and Copilot Business for non-GHE - Understand Copilot for non-GitHub customers - Define GitHub Copilot in the IDE - Define GitHub Copilot Chat in the IDE - Describe the different ways to trigger GitHub Copilot (chat, inline chat, suggestions, multiple suggestions, exception handling, CLI)
Identify the main features with GitHub Copilot Individual	- Explain the difference between GitHub Copilot Individual and GitHub Copilot Business (data exclusions, IP indemnity, billing, etc.) - Understand the available features in the IDE for GitHub Copilot Individual
Identify the main features of GitHub Copilot Business	- Demonstrate how to exclude specific files from GitHub Copilot - Demonstrate how to establish organization-wide policy management

Topic	Details
	- Describe the purpose of organization audit logs for GitHub Copilot Business - Explain how to search audit log events for GitHub Copilot Business - Explain how to manage GitHub Copilot Business subscriptions via the REST API
Identify the main features with GitHub Copilot Chat	- Identify the use cases where GitHub Copilot Chat is most effective - Explain how to improve performance for GitHub Copilot Chat - Identify the limitations of using GitHub Copilot Chat - Identify the available options for using code suggestions from GitHub Copilot Chat - Explain how to share feedback about GitHub Copilot Chat - Identify the common best practices for using GitHub Copilot Chat - Identify the available slash commands when using GitHub Copilot Chat
Identify the main features with GitHub Copilot Enterprise	- Explain the benefits of using GitHub Copilot Chat on GitHub.com - Explain GitHub Copilot pull request summaries - Explain how to configure and use Knowledge Bases within GitHub Copilot Enterprise - Describe the different types of knowledge that can be stored in a Knowledge Base (e.g., code snippets, best practices, design patterns) - Explain the benefits of using Knowledge Bases for code completion and review (e.g., improve code quality, consistency, and efficiency) - Describe instructions for creating, managing, and searching Knowledge Bases within GitHub Copilot Enterprise, including details on indexing

Topic	Details
	and other relevant configuration steps - Explain the benefits of using custom models
Using GitHub Copilot in the CLI	- Discuss the steps for installing GitHub Copilot in the CLI - Identify the common commands when using GitHub Copilot in the CLI - Identify the multiple settings you can configure within GitHub Copilot in the CLI
How GitHub Copilot works and handles data (15%)	
Describe the data pipeline lifecycle of GitHub Copilot code suggestions in the IDE	- Visualize the lifecycle of a GitHub Copilot code suggestion - Explain how GitHub Copilot gathers context - Explain how GitHub Copilot builds a prompt - Describe the proxy service and the filters each prompt goes through - Describe how the large language model produces its response - Explain the post-processing of GitHub Copilot’s responses through the proxy server - Identify how GitHub Copilot identifies matching code
Describe how GitHub Copilot handles data	- Describe how the data in GitHub Copilot individual is used and shared - Explain the data flow for GitHub Copilot code completion - Explain the data flow for GitHub Copilot Chat - Describe the different types of input processing for GitHub Copilot Chat (types of prompts it was designed for)
Describe the limitations of GitHub Copilot (and LLMs in general)	- Describe the effect of most seen examples on the source data - Describe the age of code suggestions (how old and relevant the data is) - Describe the nature of GitHub Copilot providing reasoning and context from a prompt

Topic	Details
	vs calculations - Describe limited context windows
Prompt Crafting and Prompt Engineering (9%)	
Describe the fundamentals of prompt crafting	- Describe how the context for the prompt is determined - Describe the language options for promoting GitHub Copilot - Describe the different parts of a prompt - Describe the role of prompting - Describe the difference between zero-shot and few-shot prompting - Describe the way chat history is used with GitHub Copilot - Identify prompt crafting best practices when using GitHub Copilot
Describe the fundamentals of prompt engineering	- Explain prompt engineering principles, training methods, and best practices - Describe the prompt process flow
Developer use cases for AI (14%)	
Improve developer productivity	- Describe how AI can improve common use cases for developer productivity - Learning new programming languages and frameworks - Language translation - Context switching - Writing documentation - Personalized context-aware responses - Generating sample data - Modernizing legacy applications - Debugging code - Data science - Code refactoring - Discuss how GitHub Copilot can help with SDLC (Software Development Lifecycle) management

Topic	Details
	- Describe the limitations of using GitHub Copilot - Describe how to use the productivity API to see how GitHub Copilot impacts coding
Testing with GitHub Copilot (9%)	
Describe the options for generating testing for your code	- Describe how GitHub Copilot can be used to add unit tests, integration tests, and other test types to your code - Explain how GitHub Copilot can assist in identifying edge cases and suggesting tests to address them
Describe the different SKUs for GitHub Copilot	- Describe the different SKUs and the privacy considerations for GitHub Copilot - Describe the different code suggestion configuration options on the organization level - Describe the GitHub Copilot Editor config file
Privacy fundamentals and context exclusions (15%)	
Enhance code quality through testing	- Describe how to improve the effectiveness of existing tests with GitHub Copilot's suggestions - Describe how to generate boilerplate code for various test types using GitHub Copilot - Explain how GitHub Copilot can help write assertions for different testing scenarios
Leverage GitHub Copilot for security and performance	- Describe how GitHub Copilot can learn from existing tests to suggest improvements and identify potential issues in the code - Explain how to use GitHub Copilot Enterprise for collaborative code reviews, leveraging security best practices, and performance considerations - Explain how GitHub Copilot can identify potential security vulnerabilities in your code - Describe how GitHub Copilot can suggest code optimizations for improved performance

Topic	Details
Identify content exclusions	- Describe how to configure content exclusions in a repository and organization - Explain the effects of content exclusions - Explain the limitations of content exclusions - Describe the ownership of GitHub Copilot outputs
Safeguards	- Describe the duplication detector filter - Explain contractual protection - Explain how to configure GitHub Copilot settings on GitHub.com - Enabling/disabling duplication detection - Enabling/disabling prompt and suggestion collection - Describe security checks and warnings
Troubleshooting	- Explain how to solve the issue if code suggestions are not showing in your editor for some files - Explain why context exclusions may not be applied - Explain how to trigger GitHub Copilot when suggestions are either absent or not ideal - Explain steps for context exclusions in code editors

Microsoft GH-300 Sample Questions:

Question: 1

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

a) They specify the scope that GitHub Copilot should examine and the goal to be achieved.
b) They determine the color scheme used by GitHub Copilot Chat.
c) They control the volume of the audio output from GitHub Copilot Chat.
d) They influence the programming language used for code suggestions.

Answer: a

Question: 2

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

a) By providing code suggestions based on open-source libraries only
b) By suggesting code without considering the project context
c) By randomly generating code snippets
d) By tailoring coding assistance, answering questions, and suggesting code aligned with the organization's standards and best practices

Answer: d

Question: 3

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

a) Based on the context of code in the editor.
b) Based on the programming language used.
c) Based on the length of the code written.
d) Based on a random selection of popular coding patterns.

Answer: a

Question: 4

In the outbound flow of GitHub Copilot, which of the following actions might occur after the code suggestion is generated?

a) The suggestion is presented to the user for review and acceptance
b) The suggestion is immediately integrated into the user's code
c) The suggestion is sent to a third-party server for evaluation
d) The suggestion is automatically deleted to protect user privacy

Answer: a

Question: 5

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

a) It provides a user interface for manually writing tests.
b) It automatically runs the tests and provides the results.
c) It suggests completions and generates tests based on the code context.
d) It creates a full test suite without any user input.

Answer: c

Question: 6

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?

a) Policies
b) Your organizations in your profile dropdown menu
c) Settings in your profile dropdown menu
d) Selected teams/users

Answer: b

Question: 7

What is the primary purpose of the toxicity filter in GitHub Copilot?

a) To ensure code suggestions are syntactically correct
b) To prevent the generation of code that violates intellectual property rights
c) To eliminate harmful or offensive content in code suggestions
d) To suggest code that adheres to specific coding standards

Answer: c

Question: 8

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

a) To make the code more complex and challenging for other developers.
b) To create a more readable and maintainable codebase that's easier for other developers to understand and work with.
c) To increase the size of the codebase.
d) To showcase the developer's coding skills.

Answer: b

Question: 9

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

a) The number of lines of code in the program.
b) The time required to write code.
c) The number of developers working on the project.
d) Readability, complexity, modularity, reusability, testability, extensibility, reliability, performance, security, scalability, usability, and portability.

Answer: d

Question: 10

Which of the following is NOT a principle of effective prompt engineering for GitHub Copilot?

a) Clarity - Focus on a single, well-defined task.
b) Verbosity - Provide extensive and detailed descriptions.
c) Specificity - Use clear and explicit instructions.
d) Surround - Utilize descriptive filenames and keep related files open.

Answer: b

Study Guide to Crack Microsoft GitHub Copilot GH-300 Exam:

- Getting details of the GH-300 syllabus, is the first step of a study plan. This pdf is going to be of ultimate help. Completion of the syllabus is must to pass the GH-300 exam.
- Making a schedule is vital. A structured method of preparation leads to success. A candidate must plan his schedule and follow it rigorously to attain success.
- Joining the Microsoft provided training for GH-300 exam could be of much help. If there is specific training for the exam, you can discover it from the link above.
- Read from the GH-300 sample questions to gain your idea about the actual exam questions. In this PDF useful sample questions are provided to make your exam preparation easy.
- Practicing on GH-300 practice tests is must. Continuous practice will make you an expert in all syllabus areas.

Reliable Online Practice Test for GH-300 Certification

Make EduSum.com your best friend during your Microsoft GitHub Copilot exam preparation. We provide authentic practice tests for the GH-300 exam. Experts design these online practice tests, so we can offer you an exclusive experience of taking the actual GH-300 exam. We guarantee you 100% success in your first exam attempt if you continue practicing regularly. Don’t bother if you don’t get 100% marks in initial practice exam attempts. Just utilize the result section to know your strengths and weaknesses and prepare according to that until you get 100% with our practice tests. Our evaluation makes you confident, and you can score high in the GH-300 exam.

Start Online practice of GH-300 Exam by visiting URL
https://www.edusum.com/microsoft/gh-300-microsoft-github-copilot