



**Free Questions for [GitHub-Foundations](#) by [vceexamstest](#)**

**Shared by [Pace](#) on [12-08-2024](#)**

**For More Free Questions and Preparation Resources**

**[Check the Links on Last Page](#)**

# Question 1

---

## Question Type: MultipleChoice

---

What features are offered by Copilot for Business that are not offered by Copilot for individuals?

(Each answer presents a complete solution. Choose three.)

### Options:

---

- A- Offers multi-line function suggestions
- B- Organization-wide policy management
- C- Blocks suggestions matching public code
- D- VPN proxy support via self-signed certificates
- E- Support for organization or enterprise GitHub accounts
- F- Plugs directly into the editor

### Answer:

---

B, C, E

## **Explanation:**

---

GitHub Copilot for Business offers several features that are tailored to the needs of organizations, providing more control, security, and support compared to the individual version.

Organization-wide Policy Management:

Option B is correct because Copilot for Business allows organizations to manage policies across their entire user base, providing control over how Copilot is used within the organization.

Blocking Suggestions Matching Public Code:

Option C is correct because Copilot for Business includes enhanced security features, such as blocking code suggestions that match public code to prevent inadvertent use of unlicensed code.

Support for Organization or Enterprise GitHub Accounts:

Option E is correct because Copilot for Business supports integration with GitHub Enterprise accounts, offering additional administrative controls and integration capabilities.

Incorrect Options:

Option A (multi-line function suggestions) is a feature available in both individual and business versions, so it does not differentiate the business offering.

Option D (VPN proxy support via self-signed certificates) is not a primary differentiator of Copilot for Business.

Option F (plugs directly into the editor) is true for both individual and business versions, so it is not unique to Copilot for Business.

## Question 2

---

**Question Type:** MultipleChoice

---

The difference between GitHub Desktop and github.com is that Desktop:

**Options:**

---

- A- Is a standalone software application.
- B- Enables integration with office suite applications.
- C- Is only available on Windows operating systems.
- D- Offers a graphical user interface.
- E- Is a self-hosted version of GitHub.

**Answer:**

---

D

## Explanation:

---

GitHub Desktop is a standalone application that provides a graphical user interface (GUI) for interacting with GitHub repositories, as opposed to the command-line or web-based interfaces available on github.com.

Graphical User Interface:

Option D is correct because GitHub Desktop offers a GUI, making it easier for users to manage repositories, perform commits, and handle other Git-related tasks without needing to use the command line.

Incorrect Options:

Option A is partially correct in that GitHub Desktop is a standalone application, but the key difference is the GUI.

Option B is incorrect because GitHub Desktop does not specifically enable integration with office suite applications.

Option C is incorrect because GitHub Desktop is available on both Windows and macOS.

Option E is incorrect because GitHub Desktop is not a self-hosted version of GitHub; it is a client application for accessing GitHub repositories.

[GitHub Docs: GitHub Desktop Documentation](#)

## Question 3

---

**Question Type: MultipleChoice**

---

If there are multiple README files, which of the following locations will be displayed first?

**Options:**

---

A- .github

B- /src

C- Root

D- /docs

**Answer:**

---

C

**Explanation:**

---

When multiple README files exist in different locations within a GitHub repository, the README.md file located in the root directory of the repository will be displayed first by default. This file serves as the main documentation for the repository and is automatically rendered on the repository's home page.

Root Directory:

Option C is correct because the README.md file in the root directory is prioritized and displayed first on GitHub. This is the standard behavior for how GitHub presents documentation.

Incorrect Options:

Option A (.github) is incorrect because while a README.md file in the .github directory might be used for certain configurations, it is not the first to be displayed.

Option B (/src) is incorrect because the README.md in the src directory is not prioritized over the root.

Option D (/docs) is incorrect because documentation in the /docs folder is typically secondary to the root README.md.

[GitHub Docs: About READMEs](#)

## Question 4

---

**Question Type:** MultipleChoice

---

What are primary benefits of using GitHub issues templates?

(Each answer presents a complete solution. Choose two.)

## Options:

---

- A- To automatically label or assign newly created issues
- B- To provide an easy-to-fill-out form for creating new issues
- C- To easily coerce existing issues into a standard format
- D- To automatically create new branches when issues are created

## Answer:

---

A, B

## Explanation:

---

The primary benefits of using GitHub issues templates include:

A . To automatically label or assign newly created issues: Issue templates can be configured to automatically apply labels or assign users when the issue is created, helping to streamline triage and management processes.

B . To provide an easy-to-fill-out form for creating new issues: Templates provide a standardized format for submitting issues, ensuring that all necessary information is captured and reducing the need for follow-up questions.

Coercing existing issues into a standard format or automatically creating new branches when issues are created are not functions provided by GitHub issues templates.



## Question 5

---

**Question Type:** MultipleChoice

---

Which of the following steps are part of the Codespaces lifecycle?

(Each answer presents a complete solution. Choose three.)

### Options:

---

- A- Commit
- B- Clone
- C- Rebuild
- D- Rollback
- E- Delete
- F- Create
- G- Install

### Answer:

---

C, E, F

## **Explanation:**

---

The Codespaces lifecycle on GitHub includes several key steps:

Create: This is the step where a new Codespace is initiated.

Rebuild: A Codespace can be rebuilt to ensure that the environment is up-to-date with the latest code or configurations.

Delete: Once a Codespace is no longer needed, it can be deleted to free up resources.

Committing, cloning, or installing are typical Git operations but are not considered part of the specific lifecycle steps for a GitHub Codespace.

## **Question 6**

---

**Question Type:** MultipleChoice

---

From the list of projects for an organization, how can a user identify a GitHub Projects template?

**Options:**

---

- A- Check the 'show template' checkbox.
- B- Use the 'is' filter in the search text box.
- C- Select the Templates tab.
- D- View the contents in the .github/projects folder.

**Answer:**

---

C

**Explanation:**

---

In GitHub, when viewing the list of projects for an organization, a user can identify a GitHub Projects template by selecting the Templates tab. This tab specifically lists available templates that can be used to create new projects based on predefined structures and workflows.

## Question 7

---

**Question Type: MultipleChoice**

---

Which of the following options can a user do from a discussion post?

### Options:

---

- A- Duplicate the discussion
- B- Archive the discussion
- C- Create an issue from the discussion
- D- Add the discussion to README

### Answer:

---

C

### Explanation:

---

From a discussion post on GitHub, a user can Create an issue from the discussion. This feature allows users to turn a discussion into an actionable item by creating an issue directly from the discussion thread. This is particularly useful when a conversation identifies a bug, task, or enhancement that needs to be tracked in the repository.

## Question 8

---

**Question Type:** MultipleChoice

---

What is the primary purpose of creating a new branch in the GitHub flow?

**Options:**

---

- A- To create a backup of the main branch
- B- To capture information about an issue
- C- To experiment with new features or fixes
- D- To incorporate changes from a review

**Answer:**

---

C

**Explanation:**

---

In GitHub Flow, creating a new branch is a key step in the development process that allows for isolated development of new features or fixes without affecting the main codebase.

Experimenting with New Features or Fixes:

Option C is correct. The primary purpose of creating a new branch in the GitHub flow is to provide a safe space to experiment with new features or fixes. This allows developers to work on changes independently and only merge them into the main branch after they have

been reviewed and approved.

Incorrect Options:

Option A (To create a backup of the main branch) is incorrect because branches are not typically used for backups; they are for active development.

Option B (To capture information about an issue) is incorrect because issues are tracked separately; branches are for code changes.

Option D (To incorporate changes from a review) is incorrect because incorporating changes is done during the pull request process, not when creating a branch.

[GitHub Docs: GitHub Flow](#)

**To Get Premium Files for GitHub-Foundations Visit**

**<https://www.p2pexams.com/products/github-foundations>**

**For More Free Questions Visit**

**<https://www.p2pexams.com/github/pdf/github-foundations>**

