ATMIC NETWORKS
YOUR GUIDE TO GLOBAL CERTIFIATION

"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

How are commits related to pull requests?

A. Commits are made on a branch that can have a linked pull request.

B. Commits can only be made after a pull request is created.

C. Commits can only be made before a pull request is created.

D. Commits are made on a pull request that can have a linked branch.

Answer: A

Explanation:

Commits and pull requests (PRs) are fundamental concepts in Git and GitHub workflows, particularly in collaborative software development.

Commits:

Commits are individual changes or updates made to the codebase. Each commit is identified by a unique SHA-1 hash and typically includes a commit message describing the changes.

Commits are made to a specific branch in the repository. The branch could be the main branch, or more commonly, a feature branch created for specific work or a feature.

Pull Requests (PRs):

A pull request is a mechanism for developers to notify team members that a branch is ready to be merged into another branch, usually the main branch.

PRs are used to review code, discuss changes, and make improvements before the branch is merged into the target branch.

Relationship Between Commits and PRs:

Option A is correct because commits are made on a branch, and this branch can have a pull request associated with it. The pull request tracks the branch's commits and allows for code review before merging into the target branch.

Commits can be added to the branch both before and after the pull request is created. Any new commits

pushed to the branch are automatically included in the pull request.

Incorrect Options:

Option B is incorrect because commits can be made both before and after a pull request is created.

Option C is incorrect because it suggests that commits can only be made before a pull request is created, which is not true.

Option D is incorrect because commits are not made on a pull request; they are made on a branch. The pull request links a branch to another branch (e.g., feature branch to the main branch).

Reference:

GitHub Documentation: About Pull Requests

GitHub Docs: Understanding the GitHub Flow

Git Documentation: Git Basics - Getting a Git Repository

Question: 2

What is the difference between an organization member and an outside collaborator?

A. Organization base permissions do not apply to outside collaborators.

B. Two-factor authentication (2FA) is not required for outside collaborators.

C. Outside collaborators cannot be given the admin role on a repository.

D. Outside collaborators do not consume paid licenses.

Answer: A

Explanation:

In GitHub, an organization member is a user who has been added to an organization and is subject to the organization's base permissions and policies. An outside collaborator is a user who is not a member of the organization but has been granted access to one or more repositories within the organization.

Here's the difference between an organization member and an outside collaborator:

Organization Members:

Members are subject to the organization's base permissions, which apply across all repositories within the organization. These permissions might include read, write, or admin access, depending on what has been set as the default.

Members consume paid licenses if the organization is on a paid plan.

Members are required to have two-factor authentication (2FA) if the organization enforces it.

Outside Collaborators:

Outside collaborators do not have organization-wide permissions. They only have access to specific repositories to which they have been granted permission. This means organization base permissions do not apply to them (making option A correct).

Outside collaborators do not consume paid licenses. They are only counted toward the license if they are made organization members.

Outside collaborators can be granted any level of permission, including the admin role on specific repositories.

Two-factor authentication (2FA) can be enforced for outside collaborators at the repository level, depending on the organization's security settings.

Given this information, option A is the correct answer: "Organization base permissions do not apply to outside collaborators."

Reference:

GitHub Documentation: Roles in an organization

GitHub Documentation: About outside collaborators

GitHub Documentation: Managing repository access for your organization

Question: 3

What are the defining features of Git?

A. Distributed version control, open source software, and being designed for handling projects of any size with efficiency

B. Sequential version control, cloud-based hosting service, and being designed for collaboration on large projects

C. Low-cost local branching, convenient staging areas, multiple workflows, and being designed for

managing small projects

D. Centralized version control, proprietary software, and being designed for small projects

Answer: A

Explanation:

Git is a widely-used version control system that has several defining features:

Distributed Version Control:

Git is a distributed version control system, meaning that every developer has a full copy of the entire repository, including its history, on their local machine. This enables greater flexibility, as work can be done ofline and each user has access to the full project history.

Open Source Software:

Git is open-source, meaning its source code is freely available for use, modification, and distribution. This fosters a large community of users and contributors who continuously improve the software.

Efficiency with Large Projects:

Git is designed to handle projects of any size with speed and efficiency. It can manage large codebases and many contributors without significant performance degradation, making it suitable for everything from small personal projects to large, complex software systems.

Incorrect Options:

Option B is incorrect because Git is not a sequential version control system, nor is it inherently tied to cloud-based services. GitHub, GitLab, and other platforms offer cloud hosting for Git repositories, but Git itself is a version control tool.

Option C is incorrect because Git is not limited to small projects; it is designed to scale efficiently, and the other features mentioned are only partial descriptions of Git's capabilities.

Option D is incorrect because Git is not a centralized version control system; it is distributed. Additionally, Git is open-source, not proprietary, and is used for projects of all sizes.

Reference:

Pro Git Book: What is Git?

Git Documentation: Distributed Version Control

GitHub Docs: Understanding the Git Workflow

Question: 4

Who can be assigned to an Issue or pull request?
(Each answer presents a complete solution. Choose two.)

A. Anyone who has an enterprise GitHub account

B. Anyone who has commented on the Issue or pull request

C. Anyone who has a personal GitHub account

D. Anyone with write permissions to the repository

Answer: B, D

Explanation:

In GitHub, issues and pull requests (PRs) are essential tools for managing work and collaboration in a project. Assigning individuals to these issues or PRs is a way to indicate responsibility for addressing the issue or completing the PR.

Anyone with write permissions to the repository:

Users who have write permissions to a repository can be assigned to issues and pull requests. Write permissions allow users to push changes to the repository, create branches, and modify issues and pull requests. Assigning them to an issue or PR ensures they are recognized as responsible for the task.

Anyone who has commented on the Issue or pull request:

GitHub allows you to assign issues or pull requests to users who have already engaged with the discussion by commenting on it. This feature is particularly useful for quickly assigning tasks to those who are already involved in the conversation.

Incorrect Options:

Option A is incorrect because having an enterprise GitHub account alone does not necessarily grant the ability to be assigned to issues or PRs. Permission to assign is based on repository-specific roles and permissions.

Option C is incorrect because not all personal GitHub accounts can be assigned to issues or PRs. The user needs either write permissions to the repository or must have commented on the issue or PR.

Reference:

GitHub Docs: Assigning Issues and Pull Requests

GitHub Docs: Permission Levels for a Repository

This detailed explanation provides clarity on GitHub's assignment mechanics for issues and pull requests, reflecting the platform's collaborative nature.

Question: 5

Which of the following is the best GitHub feature for long-form documentation for a project?

A. Insights
B. Pull Requests
C. Projects
D. Wikis

Answer: D

Explanation:

GitHub offers a variety of features for different aspects of project management and documentation. For long-form documentation, the best feature is Wikis. Wikis in GitHub allow you to create detailed, structured documentation that is easy to navigate and edit. Each repository in GitHub can have its own Wiki, which acts as a space for collaborators to maintain project documentation, guides, manuals, or any other long-form content.

Wikis are specifically designed to host extensive documentation in a way that is easy to reference and edit over time. They support Markdown, allowing you to format your documentation effectively. Unlike the other options, Wikis are explicitly intended for the purpose of long-form content, making them the best choice for this use case.

Question: 6

Which of the following is a key characteristic of GitHub Projects?

A. Ability to visualize the commit history

B. Ability to import Gantt charts from Microsoft Project

C. Ability to create and customize multiple views

D. Ability to enforce required fields

Answer: C

Explanation:

GitHub Projects is a flexible and powerful tool for project management that allows users to manage their work with ease. One of the key characteristics of GitHub Projects is the ability to create and customize multiple views. This feature enables teams to tailor the project management experience to their specific workflow needs, offering various ways to visualize tasks, issues, and work items.

Custom Views: You can set up different views like Kanban boards, tables, or timelines, and apply filters to show only what is relevant for a particular aspect of the project. This customization allows teams to organize their work in a way that best suits their processes, making it a highly adaptable project management tool.

Other options, such as visualizing commit history (which would fall under the 'Insights' feature), importing Gantt charts (which GitHub Projects does not natively support), or enforcing required fields (which might relate to form-based tools but not to GitHub Projects specifically), do not align with the key characteristics of GitHub Projects.

Question: 7

Which of the following is an Innersource development practice?

A. Adopting open source code into the organization

B. Sharing code between teams within the organization

C. Removing open source code from the organization

D. Making all repositories publicly accessible

Answer: B

Explanation:

Innersource is a development practice where an organization adopts open-source development methodologies within its own internal environment. The primary goal of innersource is to break down silos and encourage collaboration across different teams within the organization.

Sharing Code Between Teams:

Option B is correct because innersource involves sharing code between teams within the organization, similar to how open-source communities share code across the public domain. This practice fosters collaboration, improves code quality, and allows for reuse of code, reducing duplication of efforts.

Incorrect Options:

Option A is incorrect because adopting open-source code into the organization is related to using open-source software, not specifically to innersource practices.

Option C is incorrect because removing open-source code from the organization is contrary to the principles of both open source and innersource.

Option D is incorrect because making all repositories publicly accessible refers to open source, not innersource. Innersource typically involves keeping code internal to the organization.

Reference:

GitHub Docs: What is Innersource?

Innersource Commons: The Basics

Question: 8

Which of the following GitHub syntax formats is consistent with the associated text?

A. * This is a heading

B. This is a link

C. <!-- This is a comment -->

D. This is bolded text

E. 1. This is an ordered list

Answer: C

Explanation:

GitHub supports various syntax formats that align with Markdown and HTML conventions. Here's a breakdown of the provided options:

Comment Syntax:

Option C is correct. The syntax <!-- This is a comment --> is used in Markdown files to insert comments. These comments will not be rendered in the final output, making them useful for adding notes or instructions within the code or documentation.

Incorrect Options:

Option A (* This is a heading) is incorrect because an asterisk (*) denotes an unordered list item, not a heading. A heading in Markdown is typically created with one or more hash symbols (#).

Option B (This is a link) is incorrect because this is plain text and not the syntax for creating a link. The correct syntax would be [This is a link](URL).

Option D (This is bolded text) is incorrect because this is plain text, not the correct Markdown syntax for bold text, which should be **This is bolded text** or This is bolded text .

Option E (1. This is an ordered list) is incorrect as it does represent an ordered list item, but it was not the syntax format asked about in the question. The question specifically focuses on matching associated text with syntax, where only the comment option is correct.

Reference:

GitHub Flavored Markdown (GFM)

GitHub Docs: Basic writing and formatting syntax

Question: 9

What are the key areas of focus for GitHub?
(Each answer presents a complete solution. Choose three.)

A. Nurturing a community that supports open source principles

B. Providing access and opportunities for developers

C. Providing a social media platform for project managers

D. Building a technology platform for secure code sharing and collaboration

E. Hosting video calls with other developers

Answer: A, B, D

Explanation:

GitHub focuses on several key areas that align with its mission to support developers and foster collaboration:

Nurturing a Community That Supports Open Source Principles:

Option A is correct. GitHub is a major advocate for open-source software development, providing tools and platforms that enable open collaboration. GitHub hosts millions of open-source projects and supports a community-driven approach to software development.

Providing Access and Opportunities for Developers:

Option B is correct. GitHub provides a wide range of resources, such as GitHub Education, GitHub Actions, and GitHub Marketplace, to empower developers. These tools and opportunities help developers of all levels to learn, contribute, and improve their skills.

Building a Technology Platform for Secure Code Sharing and Collaboration:

Option D is correct. GitHub's core function is to provide a platform where developers can securely share code and collaborate. Features like private repositories, branch protections, and GitHub Actions for CI/CD (Continuous Integration/Continuous Deployment) workflows highlight this focus.

Incorrect Options:

Option C is incorrect because GitHub is not a social media platform for project managers; it is a code hosting platform with social features primarily aimed at developers.

Option E is incorrect because GitHub does not focus on hosting video calls. While some integrations might allow for video conferencing, it is not a core focus of GitHub.

Reference:

GitHub Docs: The GitHub Developer Experience

GitHub Docs: About GitHub

This detailed explanation covers the primary focuses of GitHub, emphasizing its role in the opensource community and its commitment to providing a secure and collaborative platform for developers.

Question: 10

After 30 minutes of inactivity, a GitHub Codespace will:

A. Be deleted

B. Commit changes

C. Restart

D. Time out

Answer: D

Explanation:

After 30 minutes of inactivity, a GitHub Codespace will time out. This is designed to conserve resources when the Codespace is not being actively used. The session will be paused, and you'll need to reconnect to resume your work. However, the Codespace is not deleted, and any unsaved changes might not be lost but should be committed or saved to prevent data loss.

Question: 11

As a user, what feature can you use to merge proposed changes in a repository on GitHub?

A. Issues

B. Pull requests

C. Projects

D. Discussions

Answer: B

Explanation:

The feature you can use to merge proposed changes in a repository on GitHub is Pull requests. Pull requests

are a core feature of GitHub, allowing developers to propose changes to a codebase, review code, discuss the changes, and eventually merge them into the main branch. This collaborative workflow ensures that code is reviewed and vetted before becoming part of the project.

Question: 12

What layouts are available for GitHub Projects?

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

A. Roadmap

B. Kanban

C. Board

D. Table

E. Backlog

Answer: B, C, D

Explanation:

GitHub Projects supports various layouts to help teams organize and visualize their work. The available layouts include:

B . Kanban: This is a visual task management tool where tasks are represented as cards and moved across columns that represent different stages of work.

C . Board: This layout is similar to Kanban but can be more flexible, allowing users to set up boards in various ways to suit their workflow needs.

D . Table: The Table layout allows you to view your tasks in a spreadsheet-like format, making it easy to manage and edit large amounts of data at once.

Roadmap and Backlog are not standard layouts provided by GitHub Projects. While these terms might be relevant in other project management contexts, GitHub Projects specifically offers Kanban, Board, and Table layouts.

Question: 13

Which of the following describes a branch in Git?

A. A pointer to an identical snapshot of the project at a specific point in time

B. A physical copy of the entire project stored on disk

C. A separate, isolated copy of the project's codebase

D. A new repository that shares code with the original "upstream" repository

Answer: C

Explanation:

In Git, a branch is a fundamental concept that represents an independent line of development within a project. Here's a more detailed explanation:

Branch in Git:

Option C is correct because a branch in Git is essentially a separate, isolated copy of the project's codebase where you can make changes without affecting the main codebase. Branches allow developers to work on features, fixes, or experiments in parallel to the main project.

Other Options:

Option A is incorrect because while a branch does point to a specific commit (which represents a snapshot of the project), the description lacks the emphasis on the isolated and parallel development aspect that is critical to the understanding of branches.

Option B is incorrect because a branch is not a physical copy stored on disk; it is a logical reference within the repository.

Option D is incorrect because that description better fits the concept of a fork, not a branch. A fork is a new repository that is a copy of another repository, usually used to contribute back to the original ("upstream") repository.

Reference:

Git Documentation: Branches in a Nutshell

GitHub Docs: Understanding the GitHub Flow

Question: 14

Where should a repository admin navigate to view pre-built visualizations from repository data?

A. Settings

B. Issues

C. Insights

D. Charts

Answer: C

Explanation:

GitHub provides repository admins with a feature called "Insights" where they can view various prebuilt visualizations and analytics related to the repository.

Insights:

Option C is correct because the "Insights" tab in a GitHub repository offers various pre-built visualizations, including contributions, traffic, code frequency, dependency graphs, and more. This helps admins and maintainers track the project's activity and health.

Other Options:

Option A (Settings) is incorrect because the Settings tab is where you configure repository settings, permissions, and integrations, but it does not provide visualizations of repository data.

Option B (Issues) is incorrect because the Issues tab is used for tracking bugs, enhancements, and other tasks but does not provide data visualizations.

Option D (Charts) is incorrect as there is no "Charts" tab or section in GitHub. The correct location for data visualizations is under "Insights."

Reference:

GitHub Docs: Viewing Repository Insights

Question: 15

How can a user create a repository template, and what permissions are required?

A. With Admin permissions, navigate to Repository settings and select Template Repository.

B. With Maintain permissions, navigate to Organization settings, select the repository, and choose Template Repository.

C. With Admin permissions, navigate to Organization settings, select the repository, and choose Template Repository.

D. With Maintain permissions, navigate to Repository settings and select Template Repository.

Answer: A

Explanation:

Creating a repository template in GitHub requires specific steps and permissions:

Creating a Repository Template:

Option A is correct because a user with Admin permissions can navigate to the repository's settings and enable the "Template Repository" option. This allows other users to generate new repositories from this template, which includes all branches, tags, and file history.

Other Options:

Option B is incorrect because "Maintain" permissions do not allow the creation of repository templates, and the option is not found in Organization settings but in the repository settings.

Option C is incorrect because the "Template Repository" option is in the repository settings, not in Organization settings.

Option D is incorrect because "Maintain" permissions do not grant the ability to create a repository template.

Reference:

GitHub Docs: Creating a Template Repository

Question: 16

As a user, which of the following default labels is used to indicate that a maintainer needs assistance on an issue or pull request?

A. Enhancement

B. Question

C. Help wanted

D. Documentation

Answer: C

Explanation:

In GitHub, labels are used to categorize issues and pull requests, and certain default labels are provided to help manage tasks:

Help Wanted Label:

Option C is correct. The "Help wanted" label is used to indicate that the maintainer of the repository needs assistance on a particular issue or pull request. This label helps in attracting contributors who might be interested in helping with specific tasks.

Other Options:

Option A ("Enhancement") is incorrect because it indicates a request for a new feature or improvement rather than a call for help.

Option B ("Question") is incorrect because it is used to flag issues or pull requests that seek clarification or additional information, not necessarily requiring assistance.

Option D ("Documentation") is incorrect because it labels issues or PRs related to documentation, not for seeking help.

Reference:

GitHub Docs: Using Labels

Question: 17

Which of the following options is available as a default Discussion category?

A. Bug report

B. Daily check-in

C. Show and tell

D. Security concern

Answer: C

Explanation:

In GitHub Discussions, several default categories are provided to help organize conversations within a project. One of the default categories is Show and tell. This category is designed for users to showcase their work, share progress, or discuss achievements with the community. The other options listed (Bug report, Daily check-in, Security concern) are not default categories but could be custom categories created by the repository maintainers.

Question: 18

What is a benefit of using GitHub Enterprise Cloud with Enterprise Managed Users (EMU)?

A. It provides centralized control and streamlined management of user accounts through their identity provider (IdP).

B. It offers additional collaboration and content creation capabilities for managed user accounts.

C. It automatically validates user interactions using the identity provider (IdP) conditional access policy (CAP).

D. It enables GitHub user accounts access to protected resources using SAML SSO.

Answer: A

Explanation:

GitHub Enterprise Cloud with Enterprise Managed Users (EMU) integrates closely with an organization's identity provider (IdP), such as Azure Active Directory, to manage user accounts. The primary benefit of this setup is centralized control and streamlined management. It allows organizations to enforce policies, manage user permissions, and provision or deprovision accounts directly through their IdP, ensuring consistency and security across the organization. This approach is
ideal for large enterprises that require tight control over their users and resources.

Question: 19

Which of the following is the purpose of a GitHub repository?

A. To provide a folder that stores project files, including documentation, on your local machine

B. To provide a version control system designed for small projects, offering simple tools for organizing files on your laptop

C. To provide a cloud-based hosting service for project documentation, providing a secure and centralized location for file storage

D. To provide a collaborative space where developers can share and manage code files, track changes, and store revision history

Answer: D

Explanation:

A GitHub repository serves as a collaborative space where developers can share and manage code files, track changes, and store revision history. It is much more than just a folder or simple tool; it is a comprehensive version control system that allows teams to collaborate effectively on codebases. Repositories enable developers to work together, manage contributions, review code, and maintain a complete history of every change made to the project.

Question: 20

Which of the following best describes cloning a repository?

A. It creates a copy of the repository on GitHub.com.

B. It retrieves code updates from the remote repository.

C. It creates a copy of the repository on your local machine.

D. It imports your source code into a new repository.

Answer: C

Explanation:

Cloning a repository in GitHub refers to creating a copy of the repository on your local machine. This allows you to work on the project ofline, make changes, and later push those changes back to the remote repository. It does not involve creating a copy on GitHub.com (which would be forking), retrieving updates (which would be pulling), or importing source code into a new repository (which is done differently).

Question: 21

What does a CODEOWNERS file do in a repository?

A. Restricts who can edit specific files

B. Requires peer code review for code changes

C. Defines access permissions for the repository

D. Sets the reviewers for pull requests automatically

Answer: D

Explanation:

The CODEOWNERS file in a GitHub repository is used to define individuals or teams that are responsible for specific parts of the codebase. When changes are made to files or directories that match the patterns specified in the CODEOWNERS file, GitHub automatically requests reviews from the listed code owners.

Setting Reviewers Automatically:

Option D is correct because the primary purpose of a CODEOWNERS file is to automatically set reviewers for pull requests that affect the specified files or directories. This ensures that the appropriate team members are notified and review the changes before they are merged.

Incorrect Options:

Option A is incorrect because the CODEOWNERS file does not restrict who can edit specific files; it only influences who is required to review changes.

Option B is partially related but not fully accurate because while CODEOWNERS does require certain reviews, it does not mandate peer review for all code changes.

Option C is incorrect because the CODEOWNERS file does not define access permissions for the repository; it deals with code review processes.

Reference:

GitHub Docs: About CODEOWNERS

GitHub Blog: Automatically Requesting Reviews with CODEOWNERS

Question: 22

From the Organization settings, which restrictions can organization owners place on GitHub Actions usage?

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

A. Allow actions that use self-hosted runners.

B. Allow an action to be run from a Codespace.

C. Allow specified actions.

D. Allow actions by Marketplace verified creators.

E. Allow actions created by GitHub.

Answer: A, C, D

Explanation:

Organization owners on GitHub have control over how GitHub Actions can be used within their organization. They can enforce restrictions to ensure security and compliance with organizational policies.

Allow Actions That Use Self-Hosted Runners:

Option A is correct because organization owners can configure the usage of self-hosted runners, allowing greater control over the environment where actions are run.

Allow Specified Actions:

Option C is correct because organization owners can allow only specific actions to run, adding a layer of security by limiting actions to those that have been vetted.

Allow Actions by Marketplace Verified Creators:

Option D is correct because organization owners can choose to allow actions created by GitHub Marketplace verified creators, ensuring that only trusted actions are used.

Incorrect Options:

Option B is incorrect because GitHub Actions are not designed to be run directly from a Codespace; Codespaces are for development environments.

Option E is a valid choice, but since the prompt asks for only three answers, it is not included in this response.

Reference:

GitHub Docs: Managing GitHub Actions Settings for Your Organization

Question: 23

Which of the following best describes GitHub flow?

A. A branching model that uses feature branches and multiple primary branches

B. A strategy where separate branches are created for each release, and pull requests are used to collaborate on and approve releases

C. A lightweight workflow that allows for safe experimentation with new ideas and collaboration on projects through branching, pull requests, and merging

D. A strict workflow that enforces a linear development process with all changes made directly on the main branch

Answer: C

Explanation:

GitHub Flow is a simple, yet powerful, branching strategy that is widely used in modern software development. It emphasizes collaboration and flexibility.

GitHub Flow:

Option C is correct because GitHub Flow is a lightweight workflow designed for safe experimentation and collaboration. It involves creating branches for new features or fixes, opening pull requests for review, and merging changes back into the main branch after approval.

Incorrect Options:

Option A is incorrect because GitHub Flow uses a single main branch, not multiple primary branches.

Option B is incorrect because GitHub Flow is not specifically designed around releases; it is more focused on continuous development and integration.

Option D is incorrect because GitHub Flow is not strict or linear; it encourages branching and pull requests rather than direct changes on the main branch.

Reference:

GitHub Docs: Understanding the GitHub Flow

GitHub Guides: The GitHub Flow

Question: 24

Which of the following is always true about the feature preview phases Alpha and Beta?

A. Alpha features are not available to the public.

B. Alpha features are documented.

C. Alpha and Beta features offer Service Level Agreements (SLAs).

D. Beta features provide technical support.

Answer: A

Explanation:

The terms Alpha and Beta are often used in software development to describe different stages of feature testing and release.

Alpha Features:

Option A is correct because Alpha features are typically in the early stages of development and are not available to the public. They are usually tested internally or by a limited audience.

Incorrect Options:

Option B is incorrect because Alpha features are often undocumented as they are in the early development phase.

Option C is incorrect because Alpha and Beta features usually do not offer Service Level Agreements (SLAs) due to their experimental nature.

Option D is incorrect because Beta features might offer limited support, but it is not guaranteed, especially compared to fully released features.

Reference:

GitHub Docs: About Feature Previews

Question: 25

Which of the following statements most accurately describes who can access a private repository Wiki?

A. Wikis are only viewable by repository admins.

B. Wikis can be viewed by the same people who have Read access to the repository.

C. Wikis will not be visible until shared with a specific user.

D. Wikis are public regardless of whether you have access to the repository.

Answer: B

Explanation:

For private repositories on GitHub, the Wiki is accessible to anyone who has Read access to the repository. This means that if you can view the code and files in the repository, you can also view its Wiki. This makes Wikis a useful tool for documenting projects in a way that is available to all collaborators without requiring special permissions beyond those needed to access the repository itself.

Question: 26

What qualifier finds issues that mention a certain user?

A. mentions:

B. Smentioned:

C. mentioned:

D. threads:

Answer: A

Explanation:

The qualifier mentions: is used in GitHub's search functionality to find issues that mention a certain user. For example, if you want to find all issues where a specific user is mentioned, you would use mentions:username. This helps in tracking where a user has been involved in discussions across issues or pull requests.

Question: 27

Pull requests can only be created between two branches that are:

A. Authored by the same user.

B. Authored by different users.

C. The same.

D. Different.

Answer: D

Explanation:

Pull requests are created to propose changes from one branch to another. These branches must be different; otherwise, there would be no changes to propose. Typically, pull requests are made from a feature or topic branch to a main branch (such as main or master), allowing for code review and integration before the changes are merged.

Question: 28

What are three valid states for a file in a git repository?

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

A. Committed

B. Modified

C. Uncommitted

D. Staged

E. Tracked

Answer: A, B, D

Explanation:

In a Git repository, a file can be in one of the following three valid states:

Committed: The file is saved in the local repository. It is part of the permanent history of the project.

Modified: The file has been changed but not yet staged or committed. It is in the working directory.

Staged: The file has been marked to be included in the next commit. It is in the staging area, ready to be committed.

These states represent the typical lifecycle of a file as it moves through the process of being edited, reviewed, and saved in Git.

Question: 29

What is the primary purpose of creating a security policy in a repository?

A. To ensure that peer code review occurs before new changes are merged

B. To define which types of secrets are blocked with push protection

C. To describe how security vulnerabilities should be responsibly disclosed

D. To customize the repository's Dependabot configuration

Answer: C

Explanation:

The primary purpose of creating a security policy in a GitHub repository is to guide users and contributors on how to report security vulnerabilities in a responsible and secure manner. This policy outlines the preferred method of communication, timelines, and any other pertinent information related to handling security issues.

Security Policy:

Option C is correct because a security policy provides guidelines for responsibly disclosing security vulnerabilities. This helps maintainers respond to and address security concerns promptly and securely, thereby protecting the project and its users.

Incorrect Options:

Option A is incorrect because ensuring peer code review is a best practice for code quality, but it is not the primary purpose of a security policy.

Option B is incorrect because push protection for secrets is managed through repository settings, not the security policy.

Option D is incorrect because customizing Dependabot configuration is related to dependency management, not directly to security policies.

Reference:

GitHub Docs: Adding a Security Policy to Your Repository

Question: 30

What is GitHub?

A. A proprietary software platform for nurturing creativity in developers and building a technology community

B. A cloud-based hosting service for version control and collaboration, focused on creating a safe and collaborative environment for developers

C. A platform that focuses on facilitating the growth and sharing of code, specifically designed for new developers to hone their skills

D. A centralized version control system designed for nurturing a community of developers and providing access to open source projects

Answer: B

Explanation:

GitHub is a cloud-based platform that provides hosting for software development and version control using Git. It offers tools for collaboration, project management, and security to create a safe and productive environment for developers.

GitHub Overview:

Option B is correct because GitHub is primarily known as a cloud-based hosting service for Git repositories, offering a collaborative environment where developers can work together on projects, manage version control, and implement security practices.

Incorrect Options:

Option A is incorrect because GitHub is not proprietary in the sense of being closed off from version control standards; it is widely recognized as an open platform for collaboration.

Option C is incorrect because, while GitHub is accessible to new developers, it is designed for developers of all skill levels and not specifically tailored for beginners.

Option D is incorrect because GitHub is not a centralized version control system; it supports Git, which is distributed.

Reference:

GitHub Docs: About GitHub

Question: 31

Which of the following can be performed within GitHub Desktop?

A. Creating and managing issues

B. Reviewing and approving pull requests

C. Adding and cloning repositories

D. Commenting on discussions

E. Integrating with office suite software

Answer: C

Explanation:

GitHub Desktop is a graphical interface that allows users to interact with GitHub repositories. It simplifies certain Git operations without the need for command-line usage.

GitHub Desktop Capabilities:

Option C is correct because GitHub Desktop allows users to add local repositories to their GitHub account, clone repositories from GitHub to their local machine, and manage repositories effectively.

Incorrect Options:

Option A is incorrect because GitHub Desktop does not support creating or managing issues directly; this is done through the GitHub web interface.

Option B is incorrect because reviewing and approving pull requests is also managed through the GitHub web interface.

Option D is incorrect because commenting on discussions is done on the GitHub platform, not through GitHub Desktop.

Option E is incorrect because GitHub Desktop does not integrate with office suite software.

Reference:

GitHub Docs: GitHub Desktop Documentation

Question: 32

When using Organizations, GitHub Teams is better than GitHub Free because it offers:

A. Advanced tools and insights in private repositories.

B. Authentication with SAML single sign-on and increased GitHub Actions minutes.

C. Expanded storage and priority support.

D. Increased GitHub Actions minutes and additional GitHub Packages storage.

Answer: B

Explanation:

GitHub Teams, as part of GitHub's paid plans, offers additional features and capabilities compared to GitHub Free, particularly for organizations.

GitHub Teams Benefits:

Option B is correct because GitHub Teams provides advanced security features like SAML single sign- on for secure authentication, as well as increased minutes for running GitHub Actions, which are essential for continuous integration and deployment workflows.

Incorrect Options:

Option A is incorrect because private repositories and advanced tools are features available, but the key differentiator in this context is the SAML SSO and additional GitHub Actions minutes.

Option C is incorrect because while expanded storage and priority support are valuable, SAML SSO and increased GitHub Actions minutes are more central to the differences between GitHub Free and GitHub Teams.

Option D is partially correct, but since the question asks for the best reason, Option B provides the most critical features that differentiate GitHub Teams from GitHub Free.

Reference:

GitHub Docs: About GitHub Teams

Question: 33

Which of the following best describes GitHub Copilot?

A. A Visual Studio Code extension for developing AI solutions

B. An AI tool designed to replace software developers

C. An AI pair programmer that offers autocomplete-style suggestions

D. An advanced search tool to intelligently reuse existing code in your projects

Answer: C

Explanation:

GitHub Copilot is described as an AI pair programmer that offers autocomplete-style suggestions. It is a tool integrated into development environments like Visual Studio Code that helps developers by providing code suggestions as they type. Copilot can suggest entire lines or blocks of code based on the context of what you're writing, making it a valuable assistant in coding, but not a replacement for developers.

Question: 34

Where can you go to discover, browse, and install tools?

A. GitHub Marketplace

B. GitHub Apps

C. Organization settings

D. Explore

Answer: A

Explanation:

The GitHub Marketplace is the place where users can discover, browse, and install various tools and integrations that extend the functionality of GitHub. These tools can include CI/CD services, security checks, and other development utilities that enhance workflow automation and project management.

Question: 35

How can a user choose to receive ongoing updates about a specific activity on GitHub.com?

A. By automatically watching all repositories you have push access to

B. By upgrading from a free to a paid account

C. By subscribing to notifications for all activity in a repository

D. By customizing the types of notifications you will receive in the future

Answer: C

Explanation:

On GitHub, you can choose to receive ongoing updates about specific activities by subscribing to notifications for all activity in a repository. This allows you to stay informed about all changes, discussions, and updates related to a particular project. Additionally, GitHub provides the ability to customize notifications to suit your preferences.

Question: 36

Which of the following items can you customize for an individual Codespace?

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

A. Shell

B. Branch protections

C. Name

D. Default editor

E. Operating system

Answer: A, C, D

Explanation:

When using GitHub Codespaces, you can customize several aspects of the development environment:

Shell: You can choose the default shell to be used in the Codespace, such as Bash, Zsh, or PowerShell.

Name: Users can customize the name of their Codespace for easier identification.

Default editor: You can choose which editor to use within the Codespace, such as Visual Studio Code or others that may be supported.

Branch protections and the operating system are not customizable for an individual Codespace within GitHub, making the options Shell, Name, and Default editor the correct answers.

Question: 37

Which of the following are included as pre-defined repository roles?

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

A. Security

B. View

C. Triage

D. Maintain

E. Delete

F. Write

Answer: C, D, F

Explanation:

GitHub provides several pre-defined repository roles that determine the level of access and permissions a user has within a repository. The roles that are included by default are:

Triage: Allows users to manage issues and pull requests without write access to the code.

Maintain: Provides more extensive access, including managing settings, but without full administrative control.

Write: Grants permission to push changes and manage issues and pull requests.

Roles like "Security" and "Delete" are not standard pre-defined roles, and "View" is generally referred to as "Read" in GitHub's permission structure.

Question: 38

What are the two main reasons why one might fork a repository?

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

A. To create an issue or open a discussion

B. To propose changes to the base repository

C. To create a new repository based on an existing one

D. To create a new branch to develop a new feature

Answer: B, C

Explanation:

Forking a repository on GitHub is a common practice, especially when contributing to open-source projects or when you want to build on existing work. Here are the two main reasons for forking a repository:

B . To propose changes to the base repository:

One of the primary reasons for forking a repository is to make changes or improvements that you can later propose to the original repository (often called the "upstream" repository). This is typically done through a pull request. By forking the repository, you get your own copy of the project where you can freely experiment, make changes, and then propose those changes back to the original project.

C . To create a new repository based on an existing one:

Forking is also used to create a new repository that is a copy of an existing one. This allows you to work on the project independently of the original repository, effectively creating a new direction for the project or using it as a starting point for a different purpose. This is particularly useful for customization, experimentation, or when you want to build something different while still leveraging the existing codebase.

Explanation of Other Options:

A . To create an issue or open a discussion:
This is incorrect because creating an issue or opening a discussion can be done directly on the original repository without needing to fork it. Forking is unnecessary for these actions.

D . To create a new branch to develop a new feature:
While creating a new branch is related to development, it does not require a fork. Branches are typically created within the same repository to work on new features. Forking is used when you need an entirely separate copy of the repository.

Given this information, the correct answers are B and C.

Reference:

GitHub Documentation: Fork a repo

GitHub Documentation: About forks

Question: 39

Which of the following best describes GitHub Pages?

A. Webpages hosted and published through GitHub repositories

B. Handles pagination for API requests

C. Hosts long-form documentation about your project

D. Curated guides around how to use GitHub products

Answer: A

Explanation:

GitHub Pages is a feature provided by GitHub that allows you to create webpages hosted and published through GitHub repositories. It is commonly used for hosting project documentation, personal websites, or blogs directly from a GitHub repository. It integrates seamlessly with the repository, making it easy to deploy and manage website content.

Question: 40

Which of the following are displayed in the "Pinned Repositories" section of a GitHub user profile?

A. Repositories with the most recent activity
B. Repositories that were personally selected to be highlighted

C. Repositories that are owned by organizations in which the user is a member

D. Repositories with the highest number of stars

Answer: B

Explanation:

The "Pinned Repositories" section of a GitHub user profile displays repositories that were personally selected to be highlighted by the user. Users can choose which repositories they want to feature prominently on their profile, regardless of recent activity, star count, or organizational ownership.

Question: 41

Which of the following are advantages of saved replies?
(Each correct answer presents part of the solution. Choose two.)

A. Saved replies are tied to a GitHub user's personal account.

B. Saved replies are allocated at the enterprise level for all users.

C. Saved replies allow you to create a reusable response to issues and pull requests.

D. Saved replies will send auto notifications when a user is tagged to an issue.

Answer: A, C

Explanation:

Saved replies in GitHub are a feature that allows users to create and save templates of commonly used responses for issues and pull requests. This feature can significantly enhance productivity and ensure consistent communication.

Saved Replies Are Tied to a User's Personal Account:

Option A is correct because saved replies are specific to a user's GitHub account, meaning they are accessible to the user across all repositories they have access to.

Saved Replies Allow Reusable Responses:

Option C is correct because the primary purpose of saved replies is to allow users to create reusable responses for issues and pull requests, saving time and ensuring consistency.

Incorrect Options:

Option B is incorrect because saved replies are not allocated at the enterprise level; they are specific to individual user accounts.

Option D is incorrect because saved replies do not send auto notifications; they are manually inserted by the user when responding to issues or pull requests.

Reference:

GitHub Docs: Using Saved Replies

Question: 42

What are advantages of GitHub Projects over GitHub Projects Classic?

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

A. GitHub Projects has multiple layout views.

B. GitHub Projects has Insights.

C. GitHub Projects are Copilot enabled.

D. GitHub Projects can be connected to third-party tools.

Answer: A, B

Explanation:

GitHub Projects is a newer, more powerful version of project management within GitHub, offering enhanced features over the classic version.

Multiple Layout Views:

Option A is correct because GitHub Projects supports multiple views, such as board, table, and timeline views, allowing users to visualize their work in different ways according to their needs.

Insights:

Option B is correct because GitHub Projects provides insights and analytics, enabling teams to track progress and make data-driven decisions.

Incorrect Options:

Option C is incorrect because while GitHub Copilot is a tool for code suggestions, it is not directly integrated with GitHub Projects as a feature.

Option D is incorrect because both GitHub Projects and GitHub Projects Classic can be connected to third-party tools, so it is not an exclusive advantage of the newer GitHub Projects.

Reference:

GitHub Docs: Managing Projects

Question: 43

New open source contributors can receive funding from GitHub sponsors:

A. Using PayPal as a payment processor.

B. Equal to 95% of the contribution value.

C. By including GitHub matching funds.

D. After setting up a sponsored developer profile.

Answer: D

Explanation:

GitHub Sponsors allows developers and organizations to financially support open-source contributors directly on the GitHub platform.

Setting Up a Sponsored Developer Profile:

Option D is correct because before a contributor can receive funding through GitHub Sponsors, they need

Question: 44

What is the purpose of GitHub Sponsors?

A. It funds the most popular open source projects based on stars.

B. It provides a channel for GitHub to support open source projects.

C. It offers a way for companies to purchase software on GitHub.

D. It allows the developer community to financially support open source projects.

Answer: D

Explanation:

GitHub Sponsors is a program designed to provide a platform for developers and companies to financially support open-source projects and their maintainers.

Financial Support for Open Source Projects:

Option D is correct because the main purpose of GitHub Sponsors is to allow members of the developer community, including individuals and organizations, to financially support open-source projects and maintainers. This helps sustain open-source development by providing developers with the resources they need to continue their work.

Incorrect Options:

Option A is incorrect because GitHub Sponsors is not based on project popularity (e.g., stars); it is based on voluntary contributions.

Option B is incorrect because while GitHub provides the platform, the purpose of GitHub Sponsors is not for GitHub itself to fund projects, but to enable the broader community to do so.

Option C is incorrect because GitHub Sponsors is not a marketplace for purchasing software but a platform for supporting developers.

Reference:

GitHub Docs: GitHub Sponsors

Question: 45

Which of the following best describes a Codespace?

A. A lightweight editing experience that runs entirely in your browser
B. An AI pair programmer that offers autocomplete-style suggestions
C. A development environment hosted in the cloud
D. A Visual Studio Code plug-in to manage local devcontainers

Answer: C

Explanation:

A Codespace is a cloud-hosted development environment provided by GitHub. It allows developers to code in a fully-configured environment that can be accessed from any device with an internet connection. Codespaces integrate with Visual Studio Code, either through the desktop app or directly in the browser, providing a consistent development experience without the need to manually set up a local environment.

Cloud-Hosted Development Environment:

Option C is correct because GitHub Codespaces are essentially cloud-based environments that are pre-configured with the tools and dependencies needed for development. This allows developers to start coding immediately without the overhead of setting up a local environment.

Incorrect Options:

Option A is partially correct in that Codespaces can be accessed and used in a browser, but the defining feature is that it's a full development environment, not just a lightweight editor.

Option B is incorrect because this describes GitHub Copilot, not Codespaces.

Option D is incorrect because Codespaces is not a Visual Studio Code plug-in; it is a cloud-based service that integrates with Visual Studio Code.

Reference:

GitHub Docs: About GitHub Codespaces

Question: 46

While maintaining the gist history, which of the following is the most efficient way to create a public gist based on another user's gist?

A. Fork the gist.

B. Create a new gist and copy the content from the existing gist.

C. Clone the gist.

D. Request to be added to the existing gist.

Answer: A

Explanation:

Forking a gist is the most efficient way to create a public gist based on another user's gist while maintaining the history of the original gist. When you fork a gist, you create a new gist in your own account that retains a link to the original, allowing you to track changes and contribute back if desired.

Forking a Gist:

Option A is correct because forking is a straightforward way to create your own copy of another user's gist while preserving the history and making it easy to track updates.

Incorrect Options:

Option B is incorrect because creating a new gist and copying the content would not preserve the history or link back to the original gist.

Option C is incorrect because cloning is typically associated with repositories, not gists, and is more complex than forking for this purpose.

Option D is incorrect because requesting to be added to the existing gist is not a standard GitHub feature.

Reference:

GitHub Docs: Forking Gists

Question: 47

What is a gist?

A. GitHub app

B. Git repository

C. Markdown document

D. GitHub Pages site

Answer: B

Explanation:

A gist on GitHub is essentially a Git repository, albeit a simplified one, designed for sharing snippets of code, notes, or other text content. Gists support version control and can be public or secret.

Gist as a Git Repository:

Option B is correct because each gist is a Git repository, meaning you can clone it, commit changes, and even fork it, just like any other Git repository.

Incorrect Options:

Option A is incorrect because a gist is not an app; it's a feature of GitHub that provides a simplified repository.

Option C is incorrect because while a gist may contain Markdown documents, it is fundamentally a repository that can hold various file types.

Option D is incorrect because GitHub Pages is a separate service for hosting websites, not related to gists.

Reference:

GitHub Docs: About Gists

Question: 48

Which of the following can be performed within GitHub Mobile?

A. Utilizing the mobile device as a self-hosted runner

B. Managing enterprise and organization settings

C. Chat with other GitHub Mobile users via voice calling

D. Forking and cloning repositories

E. Managing notifications from github.com

Answer: E

Explanation:

GitHub Mobile provides a streamlined experience for managing your GitHub notifications and participating in discussions, but it does not offer full functionality compared to the desktop or web interface.

Managing Notifications:

Option E is correct because GitHub Mobile allows users to manage notifications, keeping them up to date with their repositories, issues, pull requests, and other activities on GitHub.

Incorrect Options:

Option A is incorrect because GitHub Mobile cannot be used as a self-hosted runner.

Option B is incorrect because managing enterprise and organization settings is not supported in GitHub Mobile.

Option C is incorrect because GitHub Mobile does not offer a chat or voice calling feature.

Option D is incorrect because forking and cloning repositories are not actions available in GitHub Mobile.

Reference:

GitHub Docs: GitHub Mobile

Question: 49

Which of the following statements most accurately describes secret gists?

A. Anyone with the URL for the gist can view the gist.

B. Secret gists require GitHub Enterprise.

C. Anyone can see the gist from the gist Discover page.

D. Users with assigned access can view the gist.

Answer: A

Explanation:

Secret gists on GitHub are "unlisted" gists, meaning they are not publicly discoverable but can be viewed by anyone who has the URL.

Visibility of Secret Gists:

Option A is correct because secret gists can be viewed by anyone who has the direct URL, making them accessible yet unlisted.

Incorrect Options:

Option B is incorrect because secret gists do not require GitHub Enterprise; they are available on all GitHub

accounts.

Option C is incorrect because secret gists do not appear on the gist Discover page.

Option D is incorrect because secret gists do not have an "assigned access" feature; access is determined by sharing the URL.

Reference:

GitHub Docs: About Gists

Question: 50

In GitHub, why is it recommended to deploy from your feature branch before merging into the main branch?

A. To directly deploy changes from the main branch without any intermediate testing

B. To speed up the process of merging changes into the main branch

C. To avoid the need for testing changes in production

D. To ensure the changes are verified and validated in a production environment

Answer: D

Explanation:

It is recommended to deploy from your feature branch before merging into the main branch to ensure the changes are verified and validated in a production environment. This practice helps in identifying any potential issues or bugs in a real-world scenario before the changes are permanently integrated into the main branch. By deploying from the feature branch, developers can catch and address issues early, reducing the risk of introducing bugs into the main branch, which is usually considered the stable branch.

Question: 51

What folder is the definition files stored in when creating custom issue forms?

A. .issues

B. .issues/ISSUE_TEMPLATE

C. .github/ISSUE_TEMPLATE

D. .GitHub

Answer: C

Explanation:

When creating custom issue forms on GitHub, the definition files are stored in the .github/ISSUE_TEMPLATE folder. This directory is used to define issue templates and forms that help standardize the information collected when users open new issues in the repository. The .github folder is a special directory used for various repository configurations and workflows.

Question: 52

What are two recommended ways of improving the discoverability of a repository?

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

A. Register the repository with GitHub search.

B. Create a README file describing the repository.

C. Add labels to categorize the repository.

D. Add topics to classify the repository.

Answer: B, D

Explanation:

Two recommended ways to improve the discoverability of a repository on GitHub are:

B . Create a README file describing the repository: A well-written README file provides essential information about the project, such as what it does, how to use it, and how to contribute. This is often the first thing potential users or contributors will see, making it critical for discoverability.

D . Add topics to classify the repository: Adding topics to your repository helps classify it under specific categories, making it easier for others to find it through GitHub's search and exploration features. Topics act

like tags, helping to connect your project with users interested in similar subjects.

Registering a repository with GitHub search and adding labels are not applicable actions for improving discoverability in the broader sense.

Question: 53

A distributed version control system is best described as a system that:

A. Relies on a central server to store the entire project history and allows developers to check out files for editing.

B. Stores project files on a cloud-based server and allows multiple developers to collaborate on the same codebase simultaneously.

C. Ensures each developer has their own local copy of the entire code repository, including the complete project history and metadata.

D. Requires developers to manually track and manage different versions of their files using naming conventions and manual backups.

Answer: C

Explanation:

A distributed version control system (DVCS) like Git is best described as a system that ensures each developer has their own local copy of the entire code repository, including the complete project history and metadata. This decentralized approach allows developers to work independently, with full access to the project's history and files, and later synchronize their changes with others. Unlike centralized systems, DVCS does not rely on a single central server, which provides greater flexibility and robustness in collaboration.

Question: 54

What best describes Markdown?

A. Markup language
B. Programming language
C. Scripting language
D. Version control system
E. Containerization solution

Answer: A

Explanation:

Markdown is a lightweight markup language with plain-text formatting syntax. It is designed to be easy to write and read in its raw form, and it can be converted into HTML and other formats. Markdown is commonly used for formatting readme files, writing messages in online discussion forums, and creating rich text documents.

Markup Language:

Option A is correct because Markdown is indeed a markup language. It is not a programming language, scripting language, version control system, or containerization solution.

Incorrect Options:

Option B is incorrect because Markdown is not a programming language; it does not involve control structures or variables.

Option C is incorrect because Markdown is not used for scripting or automation.

Option D is incorrect because Markdown does not manage version control.

Option E is incorrect because Markdown is not related to containerization technologies like Docker.

Reference:

GitHub Docs: Basic writing and formatting syntax

Question: 55

What is the minimum access needed to contribute to a repository?

A. Read

B. Triage

C. Maintain

D. Write

Answer: D

Explanation:

To contribute to a GitHub repository, a user typically needs to be able to create branches, push changes, and open pull requests. These actions require Write access, which is the minimum level of access needed to contribute code directly to a repository.

Write Access:

Option D is correct because "Write" access allows users to contribute to the repository by pushing changes, creating branches, and opening pull requests. This is the minimum required access level for contributing code.

Incorrect Options:

Option A (Read) is incorrect because "Read" access only allows viewing the repository, not making changes.

Option B (Triage) is incorrect because while Triage access allows managing issues and pull requests, it does not allow pushing code.

Option C (Maintain) is incorrect because "Maintain" access includes additional permissions beyond those needed for basic contributions, such as managing repository settings.

Reference:

GitHub Docs: Repository Roles for an Organization

Question: 56

Which of the following information is available by default in a user's GitHub profile?

A. Personal biography and profile picture

B. Public Secure Shell Protocol (SSH) keys

C. A list of the user's private repositories

D. Email address and password

Answer: A

Explanation:

A user's GitHub profile typically includes public information such as a personal biography, profile picture, and a list of public repositories. More sensitive information, like email addresses and passwords, is not publicly displayed.

Personal Biography and Profile Picture:

Option A is correct because these are standard elements displayed on a user's public GitHub profile. This information is meant to provide a brief introduction to the user and their interests or skills.

Incorrect Options:

Option B is incorrect because public SSH keys may be associated with a user's account but are not displayed by default on the profile page.

Option C is incorrect because private repositories are not listed on a public profile.

Option D is incorrect because a user's email address and password are private information and not displayed on their public profile.

Reference:

GitHub Docs: Managing Your Profile

Question: 57

GitHub Actions workflows can be directly triggered by which of the following events?

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

A. Adding a comment to a discussion post

B. Creating a new repository

C. Committing a change to a local git repository

D. Pushing to a GitHub repository

E. Disabling a GitHub runner

F. Creating an Issue

Answer: A, D, F

Explanation:

GitHub Actions are automated workflows that can be triggered by various events on GitHub. Some common events that trigger workflows include pushes to a repository, creation of issues, and comments on discussion posts.

Triggering GitHub Actions:

Option D (Pushing to a GitHub repository) is correct because this is one of the most common triggers for CI/CD workflows.

Option F (Creating an Issue) is correct because issues are commonly used as triggers for workflows, such as automatically assigning a label or notifying a team.

Option A (Adding a comment to a discussion post) is correct because actions can be triggered by activity on discussion posts, including comments.

Incorrect Options:

Option B (Creating a new repository) is incorrect because this action typically does not trigger workflows within a specific repository.

Option C (Committing a change to a local git repository) is incorrect because GitHub Actions are triggered by events on the GitHub platform, not by local commits.

Option E (Disabling a GitHub runner) is incorrect because it is related to the environment where actions are executed, not a trigger for workflows.

Reference:

GitHub Docs: Events That Trigger Workflows

Question: 58

What should be done to locate an existing action that was provided by a GitHub-approved vendor?

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

A. Create a new workflow file.

B. Search the vendor's website for a github.yaml index.

C. Confirm that the action has a verification badge.

D. Install the GitHub App that was provided by the vendor.

E. Add the vendor as an allowed Actions Source.

F. Search the GitHub Marketplace for Actions by the vendor.

Answer: C, F

Explanation:

To locate an existing GitHub Action provided by a GitHub-approved vendor, you can use the following methods:

Verification Badge:

Option C is correct because actions provided by GitHub-approved vendors will typically have a verification badge. This badge indicates that the action is from a trusted source, giving users confidence in its security and reliability.

Search the GitHub Marketplace:

Option F is correct because GitHub Marketplace is the official location to find and install actions, including those provided by third-party vendors. You can search for actions by the vendor's name to find the specific one you need.

Incorrect Options:

Option A is not necessary to locate an existing action; creating a workflow file is for implementing the action, not locating it.

Option B is incorrect because searching the vendor's website for a github.yaml index is not a standard practice for finding actions.

Option D is incorrect because installing a GitHub App is unrelated to finding an existing action.

Option E is incorrect because adding a vendor as an allowed Actions Source is a configuration step for using the action, not for locating it.

Reference:

GitHub Marketplace: Verified Actions

Question: 59

An employee needs to find all issues within organization “Avocado" containing text "404 error" and a "guacamole" label. Which of the following steps would be best to search for these results?

A. Enter query org:Avocado is:issue label:guacamole "404 error" in the search bar.

B. Go to "Avocado" organization. Select Issues under a repository. Filter issues with a "guacamole" label.

C. Enter query org:Avocado label:guacamole "404 error" in the search bar. Select "Issues" in the Filter by section.

D. Go to the Avocado organization settings. Select Repository defaults under Repository. Scroll to Repository labels and select the 'guacamole' label.

Answer: A

Explanation:

GitHub provides a powerful search syntax to filter and find specific issues across repositories in an organization.

Search Query Syntax:

Option A is correct because the query org:Avocado is:issue label:guacamole "404 error" is the best way to search for all issues within the "Avocado" organization that contain the text "404 error" and are labeled with "guacamole". This query is precise and leverages GitHub’s advanced search capabilities.

Incorrect Options:

Option B is incorrect because it requires manual filtering in a specific repository rather than searching across the entire organization.

Option C is incorrect because selecting "Issues" in the filter by section is redundant when using the query is:issue.

Option D is incorrect because accessing organization settings to look for repository labels is not relevant to

searching for issues.

Reference:

GitHub Docs: Searching Issues and Pull Requests

Question: 60

What are some scenarios that can automatically subscribe you to conversations on GitHub?
(Each answer presents a complete solution. Choose three.)

A. Pushing a commit to the default branch

B. Being added as a repo admin

C. Opening a pull request or issue

D. Commenting on a thread

E. Being assigned to an issue or pull request

Answer: C, D, E

Explanation:

On GitHub, certain actions automatically subscribe you to conversations so that you receive notifications about further activity in that thread.

Opening a Pull Request or Issue:

Option C is correct because when you open a pull request or issue, you are automatically subscribed to the conversation and will receive notifications for any updates.

Commenting on a Thread:

Option D is correct because commenting on an issue or pull request automatically subscribes you to that thread, ensuring you are notified of further comments or changes.

Being Assigned to an Issue or Pull Request:

Option E is correct because when you are assigned to an issue or pull request, you are automatically

subscribed to notifications related to it.

Incorrect Options:

Option A is incorrect because pushing a commit to the default branch does not automatically subscribe you to conversations.

Option B is incorrect because being added as a repo admin does not automatically subscribe you to specific conversations unless you engage with them.

Reference:

GitHub Docs: Subscribing to Notifications

Question: 61

Which of the following is a primary goal of GitHub's community?

A. Exclusively supporting experienced developers

B. Creating a competitive environment for developers

C. Facilitating collaboration and creativity

D. Enforcing strict code quality standards

Answer: C

Explanation:

GitHub's community is centered around enabling developers to collaborate and innovate together. The platform provides tools and environments that foster open communication, sharing of ideas, and collective problem-solving.

Facilitating Collaboration and Creativity:

Option C is correct because GitHub is designed to be a collaborative platform where developers can work together on projects, share code, and contribute to open source initiatives, all in an environment that encourages creativity.

Incorrect Options:

Option A is incorrect because GitHub is inclusive of developers of all skill levels, not just experienced ones.

Option B is incorrect because GitHub is not about creating a competitive environment; rather, it focuses on collaboration.

Option D is incorrect because while code quality is important, enforcing strict code quality standards is not the primary goal of the GitHub community.

Reference:

GitHub Docs: Building a Strong Community

Question: 62

How can a user highlight a post to the top of the Discussions page?

A. Save the discussion.

B. Pin the discussion.

C. Create an issue from the discussion.

D. Star the discussion.

Answer: B

Explanation:

To highlight a post at the top of the Discussions page on GitHub, you can Pin the discussion. Pinning a discussion ensures it remains prominently visible at the top of the list, making it easier for others to find and participate in that discussion. This is particularly useful for important announcements or frequently referenced topics.

Question: 63

As a GitHub user, where in the UI can you configure two-factor authentication (2FA) to further secure your account?

A. Profile -> Account -> 2FA

B. Repository Settings -> Secrets and Variables -> 2FA

C. Organization Settings -> Authentication Security -> 2FA

D. Settings -> Password and Authentication -> 2FA

Answer: D

Explanation:

As a GitHub user, you can configure two-factor authentication (2FA) to secure your account by navigating to Settings -> Password and Authentication -> 2FA. This section in the GitHub user interface allows you to set up and manage your 2FA methods, which provide an additional layer of security beyond just your password.

Question: 64

Which of the following two-factor authentication (2FA) methods can you use to secure a GitHub account? (Each answer presents a complete solution. Choose three.)

A. Authenticator app
B. Security questions

C. GitHub mobile

D. Security keys

E. Single sign-on

Answer: A, C, D

Explanation:

The following two-factor authentication (2FA) methods can be used to secure a GitHub account:

A . Authenticator app: You can use an authenticator app (like Google Authenticator or Authy) to generate time-based one-time passwords (TOTP) for logging in.

C . GitHub mobile: The GitHub mobile app can also be used to receive 2FA codes, adding convenience for users who prefer to manage everything from their mobile devices.

D . Security keys: Physical security keys (such as YubiKeys) can be used as a strong form of 2FA, requiring physical access to the key to authenticate.

Security questions and Single sign-on (SSO) are not considered 2FA methods in the context of GitHub account security.

Question: 65

Which of the following are available statuses of a pull request?

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

A. Draft

B. Closed

C. Rebasing

D. Merged

E. Modified

F. Open

Answer: A, B, D, F

Explanation:

Pull requests (PRs) on GitHub can have several statuses that indicate their current state in the development and review process:

Draft:

Option A is correct. A pull request can be in a "Draft" status, indicating that it is a work in progress and not yet ready for review.

Closed:

Option B is correct. A pull request can be "Closed" without being merged, which might happen if the proposed changes are not needed or are incorporated differently.

Merged:

Option D is correct. A pull request that has been reviewed and approved can be "Merged" into the target branch, indicating that the changes have been successfully incorporated.

Open:

Option F is correct. An "Open" pull request is one that is active and awaiting review or further action.

Incorrect Options:

Option C (Rebasing) is incorrect because "Rebasing" is not a status; it's an operation that can be performed on branches.

Option E (Modified) is incorrect because there is no "Modified" status for pull requests.

Reference:

GitHub Docs: About Pull Requests

Question: 66

Workflows can reference actions in:

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

A. Any public repository.

B. The same repository as your workflow file.

C. GitHub Packages.

D. An enterprise marketplace.

E. A published Docker container image on Docker Hub.

Answer: A, B, E

Explanation:

In GitHub Actions workflows, actions can be referenced from various sources depending on the needs of the workflow.

Any Public Repository:

Option A is correct. Actions can be referenced from any public GitHub repository, allowing the reuse of shared actions across multiple projects.

The Same Repository as Your Workflow File:

Option B is correct. Actions stored in the same repository as the workflow file can be referenced directly, which is common for custom actions specific to that project.

A Published Docker Container Image on Docker Hub:

Option E is correct. Workflows can reference actions that are provided as Docker container images hosted on Docker Hub, allowing integration of complex tools and environments.

Incorrect Options:

Option C (GitHub Packages) is incorrect as it is more commonly used for storing and managing dependencies, not actions.

Option D (An enterprise marketplace) is incorrect because GitHub Actions are not directly referenced from an enterprise marketplace but rather from public repositories or Docker images.

Reference:

GitHub Docs: Reusing Workflows

Question: 67

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

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.

Reference:

GitHub Docs: GitHub Flow

Question: 68

Workflows can reference actions in:
(Each correct answer presents a complete solution. Choose three.)

A. Any public repository.

B. The same repository as your workflow file.

C. GitHub Packages.

D. An enterprise marketplace.

E. A published Docker container image on Docker Hub.

Answer: A, B, E

Explanation:

As mentioned in the answer to Question no. 66, GitHub Actions workflows can reference actions from a variety of sources:

Any Public Repository:

Option A is correct. Actions can be sourced from any public GitHub repository.

The Same Repository as Your Workflow File:

Option B is correct. Actions within the same repository as the workflow file can be referenced directly.

A Published Docker Container Image on Docker Hub:

Option E is correct. Workflows can also use actions provided as Docker container images from Docker Hub.

Incorrect Options:

Option C and D are not relevant for directly referencing actions in workflows.

Reference:

GitHub Docs: Reusing Workflows

Question: 69

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

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: 70

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

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: 71

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

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

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: 72

What are primary benefits of using GitHub issues templates?

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

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: 73

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

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.

Reference:

GitHub Docs: About README

Question: 74

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

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.

Reference:

GitHub Docs: GitHub Desktop Documentation

Question: 75

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

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

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.

Reference:

GitHub Docs: GitHub Copilot for Business