Interview Questions:

1. What is Git, and why is it widely used in software development?

Answer: Git is a distributed version control system that allows multiple developers to collaborate on a project. It tracks changes to files, enables branching and merging, and provides a complete history of code changes. Git is popular due to its speed, scalability, and ability to handle both small and large projects efficiently.

2. What are the main differences between a centralized version control system and a distributed version control system like Git?

Answer: In a centralized version control system, the codebase is stored in a central server, and developers need to connect to the server to access or commit changes. In contrast, Git is a distributed version control system where each developer has a local copy of the entire repository. This enables offline work, faster operations, and easier collaboration.

3. How does Git handle merge conflicts, and what are some strategies to resolve them?

Answer: Merge conflicts occur when Git is unable to automatically merge changes from different branches. Git marks the conflicting sections in the affected files, and it is the developer's responsibility to resolve the conflicts manually. Strategies to resolve conflicts include reviewing the conflicting changes, modifying the code to ensure

compatibility, and communicating with other team members to reach a consensus.

4. What is the purpose of Git stash, and how is it used?

Answer: Git stash allows developers to save their changes without committing them to the repository. It is useful when developers need to switch to a different branch or work on an unrelated task temporarily. Stashing allows them to save their changes and revert the working directory to the previous state. Later, the changes can be applied back to the working directory using the stash.

5. Explain the difference between "git revert" and "git reset" commands in Git.

Answer: "git revert" is used to create a new commit that undoes the changes made in a previous commit. It is a safe way to undo changes because it preserves the commit history. On the other hand, "git reset" is used to move the current branch pointer to a different commit. It can be used to discard commits, remove commits from the history, or reset the staging area. It should be used with caution as it modifies the commit history.

6. How does Git handle collaboration between multiple developers on the same project?

Answer: Git allows multiple developers to work on the same project simultaneously by providing branching and merging capabilities. Each developer can create their own branch to work on specific features or fixes. Once the changes are tested and reviewed, they can be merged into

the main branch using a pull request or merge operation. Git ensures that the changes from different branches are integrated correctly while preserving the commit history.

7. What is the purpose of a Git remote repository, and how is it different from a local repository?

Answer: A Git remote repository is a repository hosted on a server that serves as a centralized location for storing the codebase. It allows collaboration among multiple developers by providing a common repository for pushing and pulling changes. A local repository, on the other hand, is a copy of the remote repository on a developer's machine. Local repositories enable offline work and provide a sandbox environment for making changes before pushing them to the remote repository.

8. How can you revert a commit in Git, and what are the potential consequences?

Answer: To revert a commit in Git, the "git revert" command is used. It creates a new commit that undoes the changes introduced by the specified commit. The original commit remains in the history, and the revert commit is added as a new commit. It is essential to consider the consequences of reverting a commit, as it can impact the project's history and potentially introduce conflicts or inconsistencies if other changes depend on the reverted commit.

9. What is Git bisect, and how is it used in debugging?

Answer: Git bisect is a command used to perform a binary search through the commit history to identify the commit that introduced a bug or regression. It allows developers to efficiently locate the problematic commit by performing a series of tests. By marking specific commits as good or bad, Git bisect narrows down the range of commits to search and helps identify the commit responsible for the issue.

10. How can Git tags be used in the versioning of software releases?

Answer: Git tags are used to mark specific commits as important milestones or releases in the project's history. They provide a way to reference specific versions of the codebase easily. Tags are often used to mark stable releases, major releases, or specific points in the development cycle. They help in maintaining a clear history and enable developers to easily switch between different versions of the code.