**Code Versioning Tools (GIT, TFS, SVN, Azure DevOps)**

1. **Q:** How would you recover a deleted branch in Git without using the reflog?  
   **A:** Use git fsck to find the dangling commit and recreate the branch.
2. **Q:** What is the difference between a shallow clone and a deep clone in Git?  
   **A:** Shallow clone fetches only the latest commit, while deep clone fetches the entire history.
3. **Q:** How would you resolve a merge conflict in a large binary file?  
   **A:** Use a custom merge tool or manually choose the correct version.
4. **Q:** What is the difference between git rebase and git merge? When would you use one over the other?  
   **A:** Rebase rewrites commit history, while merge preserves it. Use rebase for cleaner history and merge for collaborative workflows.
5. **Q:** How would you implement a Git hook to enforce coding standards?  
   **A:** Use a pre-commit hook to run linting tools and reject non-compliant code.
6. **Q:** What is the difference between git reset and git revert?  
   **A:** Reset moves the branch pointer, while revert creates a new commit to undo changes.
7. **Q:** How would you squash multiple commits into a single commit without using interactive rebase?  
   **A:** Use git merge --squash to combine commits into a single one.
8. **Q:** What is the difference between git fetch and git pull?  
   **A:** Fetch retrieves remote changes, while pull fetches and merges them.
9. **Q:** How would you implement a Git workflow for a large team with multiple feature branches?  
   **A:** Use GitFlow or a similar branching model with develop, feature, and release branches.
10. **Q:** What is the difference between git stash and git cherry-pick?  
    **A:** Stash temporarily saves changes, while cherry-pick applies specific commits to another branch.
11. **Q:** How would you implement a Git bisect to find a bug in a large codebase?  
    **A:** Use git bisect to perform a binary search between a known good and bad commit.
12. **Q:** What is the difference between git submodule and git subtree?  
    **A:** Submodule links external repos, while subtree merges them into the main repo.
13. **Q:** How would you implement a Git hook to prevent pushing to the main branch?  
    **A:** Use a pre-push hook to check the branch name and reject pushes to main.
14. **Q:** What is the difference between git reflog and git log?  
    **A:** Reflog tracks all reference changes, while log shows commit history.
15. **Q:** How would you implement a Git workflow for a monorepo?  
    **A:** Use tools like lerna or nx to manage multiple projects in a single repo.
16. **Q:** What is the difference between git blame and git log?  
    **A:** Blame shows who last modified each line, while log shows commit history.
17. **Q:** How would you implement a Git hook to enforce commit message conventions?  
    **A:** Use a commit-msg hook to validate message format.
18. **Q:** What is the difference between git cherry-pick and git rebase?  
    **A:** Cherry-pick applies specific commits, while rebase rewrites commit history.
19. **Q:** How would you implement a Git workflow for a distributed team?  
    **A:** Use feature branches, pull requests, and regular syncs with the main branch.
20. **Q:** What is the difference between git tag and git branch?  
    **A:** Tag marks a specific point in history, while branch is a movable pointer to commits.