Assignment

What is Git?

Git is a distributed version control system used for tracking changes in source code during software development. It allows multiple developers to collaborate on a project, maintain different versions, and merge changes efficiently.

Explain the difference between Git and GitHub.

Git is a version control system that manages and tracks changes in source code, while GitHub is a web-based platform that provides hosting and additional collaboration tools for projects using Git. GitHub serves as a platform for developers to store, share, and collaborate on Git repositories.

What is the command to initialize a Git repository?

The command to initialize a Git repository is:

```
git init
```

What is the purpose of 'git status' command in Git?

The 'git status' command in Git is used to show the current status of the working directory and staging area. It displays information about modified,

untracked, and staged files, helping users understand the state of their repository.

Explain the concept of branching in Git.

Branching in Git allows for the creation of independent lines of development. It enables developers to work on features or fixes in isolation, keeping the main codebase unaffected until changes are ready to be merged. Branches can be created, switched, merged, and deleted, providing a flexible and organized way to manage code changes.

How do you revert a commit that has already been pushed and made public?

To revert a commit that has already been pushed and made public, you can use the following commands:

```
git revert <commit-hash>
git push origin <branch-name>
```

This creates a new commit that undoes the changes introduced by the specified commit and then pushes the changes to the remote repository.

What is the 'git rebase' command and when would you use it?

The 'git rebase' command in Git is used to integrate changes from one branch into another by moving or combining a sequence of commits. It is often used to maintain a cleaner and linear project history. Developers use 'git rebase' to incorporate changes from a feature branch into the main branch and resolve conflicts more efficiently than traditional merging.

Explain the Git workflow in a team environment using branching and merging.

In a team Git workflow, developers create feature branches to work on specific tasks. They regularly pull changes from the main branch, make and commit their changes on their feature branches, and then push those branches to the remote repository. Once the feature is complete, a pull request is opened for code review. After approval, the feature branch is merged into the main branch. This approach helps isolate changes, enables parallel development, and maintains a clean project history.

Describe the differences between 'git merge' and 'git rebase'.

'git merge' integrates changes from one branch into another, creating a new commit and preserving the commit history of both branches. It often results in a more complex history with multiple merge commits.

'git rebase' integrates changes by moving or combining commits from one branch onto another. It creates a linear history by applying the commits individually, resulting in a cleaner and more straightforward history. However, it should be used with caution when working on shared branches, as it rewrites commit history.

Write a command to undo the last commit but leave the changes staged.

To undo the last commit but leave the changes staged, you can use the following command:

```
git reset --soft HEAD^
```

This command resets the HEAD to the previous commit, keeping the changes from the last commit in the staging area.

Explain the concept of Git hooks.

Git hooks are scripts that can be executed before or after specific Git events, such as committing, merging, or pushing. They allow developers to automate and customize their workflow by running scripts at key points in the Git process. Git hooks can be used to enforce coding standards, run tests, trigger deployment processes, and perform other tasks to maintain consistency and quality in a collaborative development environment.