GIT Exercise

(1) What is Git and why is it used?

➔ Git is a distributed version control system used for tracking changes in files and coordinating work among multiple developers. It is commonly used for source code management but can be used for any type of file-based content. Example:

(2) Explain the difference between Git pull and Git fetch.

➔ git pull: Fetches changes from the remote repository and merges them into the current branch. It is equivalent to running git fetch followed by git merge.

git pull origin master

➔ git fetch: Downloads the latest changes from the remote repository to your local repository but does not merge them into your current branch. It allows you to see what changes exist in the remote repository before merging.

git fetch origin

(3) How do you revert a commit in Git?

➔ You can revert a commit using the git revert command followed by the commit hash you want to revert. This command creates a new commit that undoes the changes made by the specific commit.

git revert <commit-hash>

(4) Describe the Git staging area.

➔ The staging area, also known as the index, is where changes to files are organized before committing them to the repository. It allows you to selectively choose which changes you want to include in the next commit.

git add <file>
git add .

(**5) What is a merge conflict, and how can it be resolved?**

➔ A merge conflict occurs when Git is unable to automatically merge changes from different branches due to conflicting modifications to the same file. To resolve a merge conflict, you need to manually edit the conflicting files to resolve the differences, then add and commit the resolved files.

git merge <branch-name>

(6) How does Git branching contribute to collaboration?

- ➔ Git branching allows multiple developers to work on different features or bug fixes simultaneously without interfering with each other's work. Each branch represents an independent line of development, and changes made in one branch can be easily merged into another.

  git branch <branch-name>
  git checkout <branch-name>

(7) What is the purpose of Git rebase?

- ➔ Git rebase is used to integrate changes from one branch into another by reapplying commits on top of the target branch. It helps maintain a linear project history and can be used to squash or rearrange commits before merging.

  git rebase <base-branch>

(8) Explain the difference between Git clone and Git fork.

git clone: Creates a copy of a repository on your local machine, including all branches and commit history.

git clone <repository-url>

git fork: Creates a copy of a repository on a remote Git server, typically on a platform like GitHub or GitLab. Forking is commonly used in open-source projects to contribute changes without direct access to the original repository.

(9) How do you delete a branch in Git?

- ➔ You can delete a branch in Git using the **git branch -d <branch-name>** command. If the branch has unmerged changes, you need to use **git branch -D <branch-name>** to force delete it.

(10) What is a Git hook, and how can it be used?

- ➔ A Git hook is a script that Git executes before or after specific actions such as committing, merging, or pushing. Git hooks can be used to automate tasks, enforce coding standards, perform tests, or integrate with external tools to enhance the development workflow. Git provides both client-side and server-side hooks.