Revision Notes: Git and GitHub Concepts

Introduction

This lesson focused on fundamental Git commands and workflows, specifically targeting Git and GitHub operations. We explored the key commands and concepts necessary for efficient version control management, integral for collaborative software development.

Agenda

1. Installation of Git and setup of GitHub student packs.
2. Commands and concepts in focus:
   - `git remote`
   - `git fetch`
   - `git pull`
   - Difference between fetch and pull
   - `git push`
   - Pull Requests
   - Forking a repository

Git Remote

What is Remote?

- A remote in Git is a common repository that all team members use to exchange their changes. It acts as a focal point for code exchange 【4:10†transcript.txt】.
- **Setup**: You can link a local repository to a remote by using `git remote add <name> <url>`. Typically, `origin` is used as a convention for the primary remote connection 【4:14†transcript.txt】.

Origin and Other Names

【4:10†transcript.txt】【4:14†transcript.txt】.

Git Fetch vs. Git Pull

Git Fetch

- `git fetch` downloads commits, files, and refs from a remote repository without integrating them into your working files. This command updates your remote-tracking branches in the background【4:3†transcript.txt】.

Git Pull

- `git pull` is a command that fetches the content from the remote repository and immediately merges it with the local repository, essentially performing both `git fetch` and `git merge` automatically【4:3†transcript.txt】.
- **Best Practice**: It is recommended to use `git pull --rebase` to ensure a cleaner, linear project history【4:18†transcript.txt】.

Git Push

- `git push` uploads your local repository content to a remote repository. Pushing is often used to publish contributions to a project【4:9†transcript.txt】.
- Before first pushing, you should set an upstream branch using `git push --set-upstream <remote> <branch>` if the branch is not tracking a remote branch【4:14†transcript.txt】【4:13†transcript.txt】.

Fork and Pull Requests

Forking a Repository

- **Fork**: A fork is a personal copy of someone else's project situated in your GitHub account. It allows you to freely make changes without affecting the original project【4:1†transcript.txt】【4:5†transcript.txt】.

- A pull request is a mechanism for a developer to notify team members that they have completed a feature. It's essentially a request to have the changes reviewed and merged into a repository【4:5†transcript.txt】.
- **Workflow**: After committing your changes in your fork, you go to the original repository and open a pull request against the base branch. If your request is accepted, your changes will be integrated into the project 【4:5†transcript.txt】.

Version Control Strategy

Read-Only and Read-Write Copies

- Your local machine maintains a read-only copy that you can use to pull changes. This helps keep your work synchronized while minimizing conflicts 【4:8†transcript.txt】.
- Changes need to be committed for them to be pushed to a remote system like GitHub. Uncommitted changes remain local until a commit is made 【4:13†transcript.txt】.

Common Errors and Tips

- **Conflicts**: When two people modify the same part of a file, a conflict occurs, which must be resolved manually【4:7†transcript.txt】.
- **Tracking Files**: Use `git add <file>` to tell Git to track changes in a file. Untracked changes do not appear in commits【4:16†transcript.txt】.

These fundamental concepts of Git and GitHub streamline collaborative coding efforts and foster efficient version control strategies, pivotal in managing software development projects. Make sure you practice these commands and workflows to embody these concepts efficiently.