Revision Notes on Git and GitHub

This document serves as comprehensive revision notes for the class on Git and GitHub, summarizing the content from the audio transcript and handwritten notes.

Introduction to Git and GitHub

Understanding Git

- **Git** is a distributed Version Control System (VCS) that allows multiple contributors to work on the same project concurrently without overwriting each other's changes【8:9†source】.
- It maintains a history of changes, enabling users to revert to any previous version of code if needed【8:9†source】.
- Being a distributed VCS, each collaborator has access to a complete version history on their local machine【8:9†source】.

Role of GitHub

- **GitHub** is a web-based platform that hosts Git repositories, allowing for centralized control and collaboration【8:9†source】.
- It serves as a remote repository that complements local repositories on developers' machines【8:12†source】【8:15†source】.

Basic Concepts in Git

Repository

- A **repository** is like a directory or storage space for your projects. It can be local (on your personal computer) or remote (hosted on a platform like GitHub)【8:15†source】.
- Local repositories are for working copies of the project, while remote repositories are shared with others for collaboration【8:19†source】.

Commits

- Making commits allows developers to save checkpoints as they develop software, aiding in the tracking of changes and the ability to go back to previous versions if necessary【8:2†source】【8:3†source】.

Branches

- Branching in Git allows multiple lines of development, enabling users to work on different features independently【8:14†source】.
- Changes in a branch can eventually be merged back into the main branch, which results in those changes becoming part of the project【8:14†source】.

Working with Git and GitHub

Basic Commands

1. **Initialize Repository**: `git init` is used to create a new Git repository from your project directory【8:18†source】.
2. **Adding Changes**: Before committing, changes need to be added to a staging area using `git add` 【8:19†source】.
3. **Committing Changes**: `git commit -m "commit message"` records changes to the repository with a message describing the changes【8:16†source】.

Forking and Cloning

- **Forking** creates a personal copy of a repository on your GitHub, allowing you to make changes independently【8:8†source】【8:13†source】.
- **Cloning** is downloading a copy of the repository to your local machine for making contributions directly【8:8†source】.

Pulling and Pushing

- **Pull**: Download the latest changes from the remote repository to your local repository【8:10†source】.
- **Push**: Upload your changes from your local repository to the remote repository【8:10†source】.

Handling Conflicts

automatically merged. They must be resolved manually【8:15†source】.

Advanced GitHub Usage

Collaboration and Contributions

1. **Pull Requests**: A pull request notifies others about the changes you have made so they can review and merge them into the main repository 【8:14†source】.
2. **Review Process**: Other contributors or repository maintainers check and merge changes into the main branch【8:14†source】.

Git for Large Projects

- For large projects, Git allows for complex workflows involving multiple branches, reviewing systems, and team collaboration tools 【8:13†source】.

These revision notes encapsulate the primary discussions and teachings of the class. Students are encouraged to practice using Git and GitHub to better understand these concepts through hands-on application.