Lesson Plan

Basic Git Command

Git

Git is a distributed version control system (DVCS) that allows multiple developers to collaborate on projects, tracking changes to files over time. It helps manage and coordinate work among developers efficiently. Here's a list of common Git commands along with examples:

1. git init

- **Usage:** Initializes a new Git repository in the current directory.

Example:

```
git init
```

This command initializes a new Git repository in the current directory, creating a hidden .git folder that Git uses to store repository information.

2. git clone

- **Usage:** Clones a remote repository into a newly created directory, creating a local copy of the repository.

Example:

```
git clone https://github.com/example/repository.git
```

- This command clones the repository located at https://github.com/example/repository.git into a new directory named repository in your current location.

3. git add

- **Usage:** Adds file contents to the index (staging area) for the next commit.

Example:

```
git add filename.txt
```

- This command stages filename.txt for the next commit. You can replace filename.txt with the actual name of the file you want to stage.

4. git commit

- **Usage:** Records changes to the repository with a message describing the changes.

Example:

```
git commit -m "Add initial version of README"
```

- This command commits the staged changes to the repository with the commit message "Add initial version of README".

5. git push
- **Usage:** Uploads local repository content to a remote repository.

Example:

```
git push origin main
```

- This command pushes the committed changes from the local main branch to the remote repository (origin).

6. git pull
- **Usage:** Fetches and merges changes from a remote repository into the current branch.

Example:

```
git pull origin main
```

- This command fetches new changes from the main branch of the remote repository (origin) and merges them into the current branch.

7. git status
- **Usage:** Displays the state of the working directory and the staging area.

Example:

```
git status
```

- This command shows which files are staged, unstaged, or untracked, and if there are any changes to be committed.

8. git log
- **Usage:** Displays the commit history of the repository.

Example:

```
git log
```

- This command shows a detailed list of commits in reverse chronological order, including commit hashes, authors, dates, and commit messages.

9. git branch
- **Usage:** Lists existing branches, creates new branches, or deletes branches.

Examples:

```
git branch
```

Lists all existing branches in the repository.

```
git branch new-feature
```

Creates a new branch named new-feature.

```
git branch -d new-feature
```

- Deletes the branch new-feature (use -D for force deletion).

10. git checkout
- **Usage:** Switches branches or restores working tree files.

Examples:

```
git checkout main
```

Switches to the main branch.

```
git checkout -b feature-branch
```

- Creates a new branch named feature-branch and switches to it (-b flag creates a new branch).

11. git merge
- Usage: Combines changes from one branch into another branch.

Example:

```
git merge feature-branch
```

- This command merges changes from feature-branch into the current branch.

12. git remote

- **Usage:** Manages connections to remote repositories.

Examples:

```
git remote add origin https://github.com/example/repository.git
```

Adds a remote named origin with the URL https://github.com/example/repository.git.

```
git remote -v
```