Git Stash:

- **Concept**: Temporarily save changes that are not ready to be committed, allowing you to switch branches or work on something else without losing your progress.
- **Usage**: `git stash`, `git stash pop`, and `git stash list`.
- **Examples**: Stashing changes before pulling the latest updates from the main branch.

Git Reset:

- **Concept**: Undo changes by moving the branch pointer to a previous commit, effectively rewriting history.
- **Usage**: `git reset --soft`, `git reset --mixed`, and `git reset --hard`.
- **Examples**: Rolling back to a previous commit while keeping or discarding changes.

Git Rebase:

- **Concept**: Reapply commits from one branch onto another, making the history linear and easier to read.
- **Usage**: `git rebase`, `git rebase --interactive`.
- **Examples**: Rewriting commit history to maintain a clean project timeline.

More Advanced Git Techniques:

- **Cherry-Picking**: Apply specific commits from one branch onto another (`git cherry-pick`).
- **Interactive Rebase**: Squash, edit, or reorder commits during a rebase.
- **Merge vs. Rebase**: When to merge and when to rebase to maintain a clear and concise commit history.
- **Git Hooks**: Automating tasks during various stages of the Git workflow.
- **Conflict Resolution**: Efficiently resolving merge conflicts that arise during complex merges or rebases.