

# Git Tasks Summary

---

## Task 8: Soft Reset Practice

Objective: Learn how to undo commits while keeping the changes staged.

### Commands Used:

```
git log
```

# View the commit history.

```
git reset --soft HEAD~2
```

# Undo the last two commits but keep the changes staged (ready to be recommitted).

```
git commit -m "New improved commit message"
```

# Create a new commit with a better commit message.

```
git log
```

# Verify the updated commit history.

## Task 9: Revert Simulation

Objective: Safely undo a commit without rewriting history.

### Commands Used:

```
git log
```

# View the commit history to get the commit hash of the wrong commit.

```
git revert <commit-hash>
```

# Create a new commit that undoes the changes made by the specific commit.

```
git log
```

# Confirm the revert commit appears in the history.

## Task 10: Stash Use Case

Objective: Temporarily save your uncommitted work without committing it.

### Commands Used:

git status

# Check current unstaged changes.

git stash

# Save all uncommitted changes to a temporary stash.

git checkout <branch-name>

# Switch to another branch.

git stash pop

# Reapply the stashed changes after returning to the branch.

git status

# Check changes after popping the stash.

### Task 11: Cherry-Pick Challenge

Objective: Apply a specific commit from one branch into another.

### Commands Used:

git checkout feature-a

# Create and switch to a new branch (feature-a).

# Make some changes and commit them

git commit -m "Add new feature"

# Commit the changes.

git checkout main

# Switch back to the main branch.

git cherry-pick <commit-hash>

# Copy a specific commit from feature-a to main.

git log --oneline

# Check that the commit now appears on both branches.