**Lab Exercise 1.1- Working with Git Revert**

**Lab Exercise: Git Revert**

This exercise will guide you through reverting changes in Git. The git revert command is used to create a new commit that undoes the changes introduced by a previous commit without modifying the history.

**Objective**

* Learn how to use git revert to undo changes from specific commits.
* Practice handling merge conflicts during a revert.

**Prerequisites**

1. Install Git on your system.
2. Set up a Git repository:

git init git-revert-lab

cd git-revert-lab

**Steps**

**1. Set Up the Repository**

1. Create a file:

echo "Line 1" > file1.txt

1. Stage and commit the file:

git add file1.txt

git commit -m "Initial commit: Add file1.txt with Line 1"

1. Add more changes:

echo "Line 1" >> file2.txt

git commit -am "Add Line 1 to file2.txt"

1. Add another change:

echo "Line 1" >> file3.txt

git commit -am "Add Line 1 to file3.txt"

1. Verify the commit history:

git log --oneline

Example output:

c3f6b6b (HEAD -> main) Add Line 1 to file3.txt

8b2a1c1 Add Line 1 to file2.txt

4d5f8e9 Initial commit: Add file1.txt with Line 1

**2. Revert the Last Commit**

1. Revert the most recent commit:

git revert HEAD

1. Git will open a text editor for the commit message. Save and close the editor to complete the revert.
2. Check the commit history:

git log --oneline

Example output:

e5d2f8a Revert "Add Line 1 to file2.txt"

c3f6b6b Add Line 1 to file3.txt

8b2a1c1 Add Line 1 to file3.txt

4d5f8e9 Initial commit: Add file1.txt with Line 1