



## Experiment – 3.3

**Student Name:** Ayush Pandey

**Branch:** CSE-DevOps

**Semester:** 4th

**Subject Name:** Git and hub

**UID:** 22BDO10038

**Section/Group-** 22BCD-1/A

**Date of Performance:** 03/04/2024

**Subject Code:** 22CSH-293

**1. Aim/Overview of the practical:** Work with your history in Git

**2. Software Used:** Git Bash.

**3. Steps for experiment/practical:**

1. Create a git repository and make a file named file.txt using vi editor.  
Make 4 commits using add and commit, while writing commit messages.

```
Ayush Pandey@Ayush MINGW64 /d/Git
$ mkdir exp10

Ayush Pandey@Ayush MINGW64 /d/Git
$ cd exp10

Ayush Pandey@Ayush MINGW64 /d/Git/exp10
$ git init
Initialized empty Git repository in D:/Git/exp10/.git/

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ vi file.txt

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git commit -m "First Commit"
[master (root-commit) 7963af2] First Commit
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ vi file.txt

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git commit -m "Second Commit"
[master 931695c] Second Commit
1 file changed, 4 insertions(+)

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ vi file.txt

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git commit -m "Third Commit"
[master 9f29cb1] Third Commit
1 file changed, 13 insertions(+)
```

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ vi file.txt

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git commit -m "Fourth Commit"
[master 185dac2] Fourth Commit
1 file changed, 2 insertions(+)
```

2. Let's perform git commit --amend to change the last commit.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git commit --amend
[master e9628f5] New Ameend Fourth Commit
Date: Tue Apr 9 22:47:38 2024 +0530
1 file changed, 2 insertions(+)
```

3. Now let's change multiple commits using git rebase while using interactive mode.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git rebase -i HEAD~3
```

This text editor window will pop up.

```
MINGW64:/d/Git/exp10
pick 931695c Second CommitOB
pick 9f29cb1 Third Commit
pick e9628f5 New Ameend Fourth Commit

# Rebase 7963af2..e9628f5 onto 7963af2 (3 commands)
#
```

Using reword to change the commit message and squash to merge that commit into above commit.

Reword will open this to edit the commit message.

```
Third Commit
```

```
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.  
#  
# Date:      Tue Apr 9 22:46:13 2024 +0530  
#  
# interactive rebase in progress; onto 7963af2
```

Squash will open this to change the commit of both commits into one to have changes being made in both the commits.

```
# This is a combination of 2 commits.  
# This is the 1st commit message:
```

```
Third Commit
```

```
# This is the commit message #2:
```

```
New Ameend Fourth Commit
```

```
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.
```

After success, this will be shown.



```
[detached HEAD e71217b] Third Commit
Date: Tue Apr 9 22:46:13 2024 +0530
1 file changed, 13 insertions(+)
[detached HEAD b725667] Third Commit
Date: Tue Apr 9 22:46:13 2024 +0530
1 file changed, 15 insertions(+)
Successfully rebased and updated refs/heads/master.

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
```

4. Splitting the commit into 2.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git rebase -i HEAD~1
```

```
edit b725667 Third Commit

# Rebase 931695c..b725667 onto 931695c (1 command)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup [-C | -c] <commit> = like "squash" but keep only the previous
```

```
Stopped at b725667... Third Commit
You can amend the commit now, with
```

```
git commit --amend
```

```
Once you are satisfied with your changes, run
```

```
git rebase --continue
```

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git reset HEAD^
Unstaged changes after reset:
M      file.txt
```

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git status
interactive rebase in progress; onto 931695c
Last command done (1 command done):
  edit b725667 Third Commit
No commands remaining.
You are currently splitting a commit while rebasing branch 'master' on '931695c'.
(Once your working directory is clean, run "git rebase --continue")

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ cat file.txt
#include <iostream>

bool isLeapYear(int year) {
    return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
}

int main() {
    int year;

    std::cout << "Enter a year: ";
    std::cin >> year;

    if (isLeapYear(year)) {
        std::cout << year << " is a leap year." << std::endl;
    } else {
        std::cout << year << " is not a leap year." << std::endl;
    }

    return 0;
}

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git commit -m "First commit after rebase"
[detached HEAD a41be3a] First commit after rebase
1 file changed, 15 insertions(+)

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ vi file.txt

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git commit -m "Second Commit after rebase"
[detached HEAD 954e252] Second Commit after rebase
1 file changed, 2 insertions(+)
```



```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master|REBASE 1/1)
$ git rebase --continue
Successfully rebased and updated refs/heads/master.

Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git log --pretty=format:"%h - %s"
954e252 - Second Commit after rebase
a41be3a - First commit after rebase
931695c - Second Commit
7963af2 - First Commit
```

Successfully splitting of commit into two.

5. Let's reorder our commits. Before that let's see the commits in current order.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git log --pretty=format:"%h - %s"
954e252 - Second Commit after rebase
a41be3a - First commit after rebase
931695c - Second Commit
7963af2 - First Commit
```



6. Using Nuclear options to delete a file from every commit using filter branch command.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git filter-branch --tree-filter 'rm -f file.txt' HEAD
WARNING: git-filter-branch has a glut of gotchas generating mangled history
rewrites. Hit Ctrl-C before proceeding to abort, then use an
alternative filtering tool such as 'git filter-repo'
(https://github.com/newren/git-filter-repo/) instead. See the
filter-branch manual page for more details; to squelch this warning,
set FILTER_BRANCH_SQUELCH_WARNING=1.
Proceeding with filter-branch...

Rewrite 7963af251c01baa1edd3a373dd97f3e21a8cee2e (1/4) (0 seconds passed, remain
Rewrite 931695c0ac1d8e9030f1557491245f97e0924900 (2/4) (1 seconds passed, remain
Rewrite a41be3ad4f9cee7a8cddacc4ccdcc3ec2c144f6a (2/4) (1 seconds passed, remain
Rewrite 954e2525e194798f39ff9022a054a035be027795 (2/4) (1 seconds passed, remain
ing 1 predicted)
Ref 'refs/heads/master' was rewritten
```

7. Changing email address globally.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp10 (master)
$ git filter-branch --commit-filter '
    if [ "$GIT_AUTHOR_EMAIL" = "Ayush@localhost" ];
    then
        GIT_AUTHOR_NAME="Ayush Pandey";
        GIT_AUTHOR_EMAIL="ayushpandey2442.com";
        git commit-tree "$@";
    else
        git commit-tree "$@";
    fi' HEAD
WARNING: git-filter-branch has a glut of gotchas generating mangled history
rewrites. Hit Ctrl-C before proceeding to abort, then use an
alternative filtering tool such as 'git filter-repo'
(https://github.com/newren/git-filter-repo/) instead. See the
filter-branch manual page for more details; to squelch this warning,
set FILTER_BRANCH_SQUELCH_WARNING=1.
Proceeding with filter-branch...

Cannot create a new backup.
A previous backup already exists in refs/original/
Force overwriting the backup with -f
```

#### 4. Result/Output/Writing Summary:

In this experiment, I learned about git amend, git rebase for changing commit messages and reordering the commits. Used git rebase to squash two commits into one and also used it to reorder the commits. Then I used nuclear options to delete file from all commits and also changing e-mail address globally.

#### Learning outcomes (What I have learnt):

1. Learnt about the git amend in commit command.
2. Learnt about changing multiple commit messages using git rebase.
3. Learnt about reordering the commits.
4. Learnt about squashing and splitting of commits.
5. Learnt about nuclear options with –filter-branch switch.

#### Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			