



## Experiment -3.3

**Student Name:** Sushant jha

**Branch:** CSE (Devops)

**Semester:** 4th

**Subject Name:** Git and Git Hub

**UID:** 22BDO10052

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

**Date of Performance:** 03/04/24

**Subject Code:** 22CSH-293

### 1. Aim/Overview of the practical:

Work with your history in Git.

### 2. Software Used:

Git Bash, Git-Hub.

### 3. Steps for experiment/practical:

1. Create a file locally and navigate to its folder using Git Bash.
2. Open and edit the file "file.txt" using the vi editor.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHub/git-9 (main)
$ git init
Initialized empty Git repository in C:/Users/ASUS TUF/Desktop/gitandgitHub/git-9
/.git/
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHub/git-9 (main)
$ vi file1.txt
```

3. Add and commit the changes to the file using Git.
4. Repeat steps 3 and 4 multiple times to create multiple commits.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHub/git-9 (main)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF
  time Git touches it

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHub/git-9 (main)
$ git commit -m "initial commit"
[main (root-commit) 8464207] initial commit
 1 file changed, 2 insertions(+)
 create mode 100644 file1.txt
```



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC  
GRADE A+  
ACCREDITED UNIVERSITY

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUB/git-9 (main)
$ git log
commit aa5c3cfe7e89a866346cd42d9fd2d015080cd0a9 (HEAD -> main)
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 02:06:55 2024 +0530

    Second commit on main

commit 38091e48bd908edf472e64bf5cf00b410e6aebfb
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 02:02:26 2024 +0530

    First commit on main

commit 8464207c54a687e84272a8e7d5d89d968737054c
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 01:35:32 2024 +0530

    initial commit
```

5. Edit the file again, stage the edit, and amend the previous commit using "git commit --amend".

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUB/git-9 (main)
$ git commit --amend
[main 8f134d6] Second commit on main now amended
Date: Wed Apr 10 02:06:55 2024 +0530
1 file changed, 1 insertion(+), 1 deletion(-)
```

6. View the commit history using "git log" to verify the changes.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUB/git-9 (main)
$ git log
commit 8f134d69b9557f62bf2267179133925c58e0bca4 (HEAD -> main)
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 02:06:55 2024 +0530

    Second commit on main now amended

commit 38091e48bd908edf472e64bf5cf00b410e6aebfb
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 02:02:26 2024 +0530

    First commit on main

commit 8464207c54a687e84272a8e7d5d89d968737054c
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 01:35:32 2024 +0530

    initial commit
```

7. Now make a new branch and go to that branch and make changes in the file using “git checkout -b feature”.



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC  
GRADE A+  
ACCREDITED UNIVERSITY

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (main)
$ git branch feature
```

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (main)
$ git checkout feature
Switched to branch 'feature'
```

8. Make several commits in this new branch.
9. Switch back to the main branch and rebase the changes from the new branch onto it.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature)
$ vi file1.txt
```

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF
```

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature)
$ git commit -m "feature branch 2nd commit"
[feature ef0a444] feature branch 2nd commit
 1 file changed, 3 insertions(+), 1 deletion(-)
```

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature)
$ git checkout main
Switched to branch 'main'
```

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (main)
$ git rebase feature
Successfully rebased and updated refs/heads/main.
```

10. Review the commit history using "git log" to analyze the changes.



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC  
GRADE A+  
ACCREDITED UNIVERSITY

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (main)
$ git log
commit ef0a444904a07c772ce7bcc987289fd97a48e20d (HEAD -> main, feature)
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 05:17:33 2024 +0530

    feature branch 2nd commit

commit c82475fcfb063f0b6e92e535c43d0f396d15fd2a
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 05:15:04 2024 +0530

    feature branch 1st commit

commit 8f134d69b9557f62bf2267179133925c58e0bca4
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 02:06:55 2024 +0530

    Second commit on main now amended

commit 38091e48bd908edf472e64bf5cf00b410e6aebfb
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 02:02:26 2024 +0530

    First commit on main

commit 8464207c54a687e84272a8e7d5d89d968737054c
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 01:35:32 2024 +0530

    initial commit
```

11.Create another new branch and make commits in it.

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (main)
$ git checkout -b feature2
Switched to a new branch 'feature2'

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature2)
$ vi file1.txt

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature2)
$ git add .

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature2)
$ git commit -m "1st commit on feature2 branch"
[feature2 05558c6] 1st commit on feature2 branch
 1 file changed, 1 insertion(+)

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgitHUb/git-9 (feature2)
$ git log
commit 00ab7fe4974361eeba4e29757d5f5f8d9bf83f20 (HEAD -> feature2)
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 05:25:25 2024 +0530

    2nd commit on feature2 branch

commit 05558c6cb957a2348e8a4696fde99a70cbeb0951
Author: sushant1236 <sushantjha1236@gmai.com>
Date:   Wed Apr 10 05:23:17 2024 +0530

    1st commit on feature2 branch
```



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.

NAAC  
GRADE A+  
ACCREDITED UNIVERSITY

12. Merge the changes from this branch into the main branch and rebase the main branch to 3

```
ASUS TUF@sushant MINGW64 ~/Desktop/gitandgithub/git-9 (feature2)
$ git checkout main
Switched to branch 'main'

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgithub/git-9 (main)
$ git merge feature2
Updating ef0a444..00ab7fe
Fast-forward
 file1.txt | 2 ++
 1 file changed, 2 insertions(+)

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgithub/git-9 (main)
$ git rebase -i HEAD~3
Successfully rebased and updated refs/heads/main.

ASUS TUF@sushant MINGW64 ~/Desktop/gitandgithub/git-9 (main)
$ git rebase -i HEAD~3
```

This will open a prompt and makes changes whatever you want to make.

```
pick ef0a444 feature branch 2nd commit
pick 05558c6 1st commit on feature2 branch
pick 00ab7fe 2nd commit on feature2 branch

# Rebase c82475f..00ab7fe onto c82475f (3 commands)
#
# 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
#                               commit's log message, unless -C is used, in which case
#                               keep only this commit's message; -c is same as -C but
#                               opens the editor
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
#           create a merge commit using the original merge commit's
#           message (or the oneline, if no original merge commit was
#           specified); use -c <commit> to reword the commit message
# u, update-ref <ref> = track a placeholder for the <ref> to be updated
#                     to this position in the new commits. The <ref> is
#                     updated at the end of the rebase
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
```



# DEPARTMENT OF ACADEMIC AFFAIRS

Discover. Learn. Empower.



13. Use "git log" to examine the updated commit history.

## Learning outcomes (What I have learnt):

1. Acquired knowledge regarding the contrast between reset and revert.
2. Gained proficiency in executing a reset operation.
3. Acquired insight into various types of rebases.
4. Obtained understanding about interactive rebases and their functionality.
5. Familiarized oneself with the commands associated with the rebase operation.

## 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.			