

Experiment – 3.2

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: Understanding the various reset modes in GitHub.

2. Software Used: Git Bash and GitHub.

3. Steps for experiment/practical:

1. Open git bash and make a directory, and change the directory. Then initialize git in it.

```
Ayush Pandey@Ayush MINGW64 /d/Git
$ mkdir exp9

Ayush Pandey@Ayush MINGW64 /d/Git
$ cd exp9

Ayush Pandey@Ayush MINGW64 /d/Git/exp9
$ git init
Initialized empty Git repository in D:/Git/exp9/.git/
```

2. Create and write in file named file.cpp

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ vi file.cpp
```

3. Perform git add and commit.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git commit -m "file.cpp created"
[master (root-commit) 31304cd] file.cpp created
1 file changed, 22 insertions(+)
create mode 100644 file.cpp
```

4. After first commit.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ cat file.cpp
#include <iostream>

int main() {
    double num1, num2;

    std::cout << "Enter the first number: ";
    std::cin >> num1;

    std::cout << "Enter the second number: ";
    std::cin >> num2;

    double sum = num1 + num2;

    std::cout << "Sum of " << num1 << " and " << num2 << "
s " << sum << std::endl;

    return 0;
}
```

5. Let's make some more changes and perform git add and commit.

```
int main() {
    double num1, num2;

    std::cout << "first number: ";
    std::cin >> num1;

    std::cout << "second number: ";
    std::cin >> num2;

    double sum = num1 + num2;

    std::cout << "Sum of " << num1 << " and " << num2 << "
is " << sum << std::endl;

    return 0;
}
~
```

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git add .

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git commit -m "Changes"
[master 21e50b6] Changes
1 file changed, 2 insertions(+), 2 deletions(-)
```

6. Now, let's use git log to see the commit history.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git log
commit 21e50b65417185bf1d00463eaf811d383fa341e3 (HEAD -> master)
Author: Ayush Pandey <ayushpandey2442@gmail.com>
Date: Tue Apr 9 21:30:50 2024 +0530

    Changes

commit 31304cd7cc86a5ae2c9aa1d73764204c23d37ae5
Author: Ayush Pandey <ayushpandey2442@gmail.com>
Date: Tue Apr 9 21:26:44 2024 +0530

    file.cpp created
```

7. We can see that in git status nothing to be committed and can see the changes in file.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git status
On branch master
nothing to commit, working tree clean

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ cat file.cpp
#include <iostream>

int main() {
    double num1, num2;

    std::cout << "first number: ";
    std::cin >> num1;

    std::cout << "second number: ";
    std::cin >> num2;

    double sum = num1 + num2;

    std::cout << "Sum of " << num1 << " and " << num2 << "
is " << sum << std::endl;

    return 0;
}
```


8. Let's perform git reset in safe mode to move back to staging area, copy the first 6-7 letters of the hash commit you want to revert back to. We can see our file in staging area.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git reset --soft 31304cd

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file.cpp
```

9. Let's perform git reset in mixed mode to go back to editing stage means move out of staging area. Same copy the last commit first 6-7 characters. Changes still remain in file.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git reset --mixed 31304cd
Unstaged changes after reset:
M       file.cpp

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git status
On branch master
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.cpp

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

We can see the changes needed to be add in staging area.

10. Let's perform the git reset in hard mode means it will move the head to commit being mentioned here in the command and all the changes are discarded.

```
Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git reset --hard 31304cd7
HEAD is now at 31304cd file.cpp created

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ git status
On branch master
nothing to commit, working tree clean

Ayush Pandey@Ayush MINGW64 /d/Git/exp9 (master)
$ cat file.cpp
#include <iostream>

int main() {
    double num1, num2;

    std::cout << "Enter the first number: ";
    std::cin >> num1;

    std::cout << "Enter the second number: ";
    std::cin >> num2;

    double sum = num1 + num2;

    std::cout << "Sum of " << num1 << " and " << num2 << "
is " << sum << std::endl;

    return 0;
}
```

4. Result/Output/Writing Summary:

In this experiment, I first created a git repository. I made two commits and used git reset commands in all three modes i.e. soft, mixed, hard and observed the changes between them and when to use which.

Learning outcomes (What I have learnt):

1. Learnt about the git reset command.
2. Learnt about the head and index in git.
3. Learnt about the differences between all the three modes of git reset – soft, mixed and hard.

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.			