MODULE -1

Managing Source Code – Git and GitHub

Submitted by:Bonagiri Manohar

Date of Submission:31-03-2025

1. L1 - Create Local git repository and demonstrate
all git reset options and revert. Compare the Differences

Step 1:we need o instatilize the Git ,By using git init command

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git init
Reinitialized existing Git repository in C:/Users/sree/Desktop/Start/Git assiessment/.git/

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 2: Create a file using touch or Vi command

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ touch q2.txt

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 3: now add the file to the staging area,by using git add command

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git add .

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 4:now we can commit the file into local repository ,by using git commit command

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git commit
On branch main
Your branch and 'origin/main' have diverged,
and have 2 and 1 different commits each, respectively.

nothing to commit, working tree clean

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 5:Git revert ,by using this command the commit we don't want we and remove it and that can be commited again with a new commit id.

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git revert 204e2e5
```

Step 6: The revert has been done with new commit id and message

```
 delete mode 100644 q2.txt

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git log --oneline
b84fb13 (HEAD -> main) revert utRevert "the file is added to local`"
a27408d the file is added to local`
438ab07 all file are commited
204e2e5 The file will be moved into stagging area
```

Step 7 :Git reset --mixed ,when we used this command the commit id which we mention after that commit id all the files are removed from local repo.

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git reset 204e2e5

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 8: Now we can see clearly the files are removed from local repo

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git status
On branch main
Your branch is behind 'origin/main' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        q1.txt

nothing added to commit but untracked files present (use "git add" to track)

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 9: Git reset -- soft "commit id" by using this command the files which are present after that commit id that will be move to staging area

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git reset -- soft 204e2e5

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 10: now we can see that all the files are moved to staging area

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git reset -- soft 204e2e5

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" if you want to integrate the remote branch with yours)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        q1.txt

nothing added to commit but untracked files present (use "git add" to track)

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 11: git reset --hard <commit id> ,by using this command the files after the commit id will be removed and the data will also be removed

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git reset -- hard 204e2e5

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Step 12: we could see the all the commits are deleted after the given coomit id

```
sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$ git log --oneline
204e2e5 (HEAD -> main) The file will be moved into stagging area

sree@DESKTOP-URR5HER MINGW64 ~/Desktop/Start/Git assiessment (main)
$
```

Differences:

Git revert :
By using this command the commit changes which we don't want we can ignore by using Git revert and after using this the new commit id and message will be displayed .
By using this Git revert the data loss will not takes place.

Git Reset :
GIt reset <commit id> or GIT mixed:
By using this commit the files can moved out local repository which are committed after the mentioned commit id .
Git reset -- soft <commit id>:
By using this we can remove the files to staging area,which are present after the mentioned commit id
Git reset -- hard <commit id>:
By using this the all files and data will be lost