**KATA 1-TASK:**

1. Use git status to see which branch you are on.
2. What does git log look like?
3. Create a file
4. What does the output from git status look like now?
5. add the file to the staging area
6. How does git status look now?
7. commit the file to the repository
8. How does git status look now?
9. Change the content of the file you created earlier
10. What does git status look like now?
11. add the file change
12. What does git status look like now?
13. Change the file again
14. Make a commit
15. What does the status look like now? The log?
16. Add and commit the newest change

**KATA 1-SOLUTION:**

The questions asked above are all output based so below is the complete set of commands for above set of questions.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

nothing to commit, working tree clean

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git log

commit f5efc15236808aa995b2906a9d550489212d2886 (HEAD -> new\_branch)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

:...skipping...

commit f5efc15236808aa995b2906a9d550489212d2886 (HEAD -> new\_branch)

commit f5efc15236808aa995b2906a9d550489212d2886 (HEAD -> new\_branch)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

Merge branch 'greeting'

commit 5d576d4a70262c3386ec3918a3e2a18f599a59b5

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:56:43 2021 +0530

grv2

commit f7d096a6d4105655231e7034cdadf323baacf6c4 (greeting)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:55:40 2021 +0530

grv1

commit 84c9c70fb7355cb4ac2e1bb95641389ee5fc4cd1 (grv)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:43:46 2021 +0530

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ nano new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

Untracked files:

(use "git add <file>..." to include in what will be committed)

new\_file

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ nano new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

Untracked files:

(use "git add <file>..." to include in what will be committed)

new\_file

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git add new\_file

warning: LF will be replaced by CRLF in git\_clone\_repo/new\_file.

The file will have its original line endings in your working directory

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git commit -m "commit-msg-1"

[new\_branch 6356d44] commit-msg-1

1 file changed, 4 insertions(+)

create mode 100644 git\_clone\_repo/new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

nothing to commit, working tree clean

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ nano new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

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: new\_file

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git add new\_file

warning: LF will be replaced by CRLF in git\_clone\_repo/new\_file.

The file will have its original line endings in your working directory

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git commit -m "commit-msg-2"

[new\_branch df5e56d] commit-msg-2

1 file changed, 1 insertion(+)

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git log

commit df5e56d1e815edd816d503563420e9d7dfeb44a4 (HEAD -> new\_branch)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:33:50 2021 +0530

commit-msg-2

commit 6356d4422133f36ffb844e2d2ac8babb1c22ab59

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:30:53 2021 +0530

commit-msg-1

commit f5efc15236808aa995b2906a9d550489212d2886

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

Merge branch 'greeting'

commit 5d576d4a70262c3386ec3918a3e2a18f599a59b5

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:56:43 2021 +0530

:

commit df5e56d1e815edd816d503563420e9d7dfeb44a4 (HEAD -> new\_branch)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:33:50 2021 +0530

commit-msg-2

commit 6356d4422133f36ffb844e2d2ac8babb1c22ab59

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:30:53 2021 +0530

commit-msg-1

commit f5efc15236808aa995b2906a9d550489212d2886

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

Merge branch 'greeting'

commit 5d576d4a70262c3386ec3918a3e2a18f599a59b5

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:56:43 2021 +0530

:

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$

**KATA 2-TASK:**

You live in your own repository. There is a file called file.txt.

1. What's the content of file.txt?
2. Overwrite the content in file.txt: echo 2 > file.txt to change the state of your file in the working directory (or sc file.txt '2' in PowerShell)
3. What does git diff tell you?
4. What does git diff --staged tell you? why is this blank?
5. Run git add file.txt to stage your changes from the working directory.
6. What does git diff tell you?
7. What does git diff --staged tell you?
8. Overwrite the content in file.txt: echo 3 > file.txt to change the state of your file in the working directory (or sc file.txt '3' in PowerShell).
9. What does git diff tell you?
10. What does git diff --staged tell you?
11. Explain what is happening
12. Run git status and observe that file.txt are present twice in the output.
13. Run git reset HEAD file.txt to unstage the change
14. What does git status tell you now?
15. Stage the change and make a commit
16. What does the log look like?
17. Overwrite the content in file.txt: echo 4 > file.txt (or sc file.txt '4' in PowerShell)
18. What is the content of file.txt?
19. What does git status tell us?
20. Run git checkout file.txt
21. What is the content of file.txt?
22. What does git status tell us?

**KATA 2-SOLUTION:**

Basically answering to 11th qsn asked, we are here coming up with the difference between the outputs of “git diff” & “git diff –staged” which is just that the former shows the difference in changes for all the files that are changed while the later shows us the difference in changes for all files that are only in staged area.

Rest all questions asked are output based and are shown below:

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ ls

new\_file zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ cat new\_file

checking

1

updating this one

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ echo 2>new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git diff

**diff --git a/git\_clone\_repo/new\_file b/git\_clone\_repo/new\_file**

**index c3e9162..e69de29 100644**

**--- a/git\_clone\_repo/new\_file**

**+++ b/git\_clone\_repo/new\_file**

@@ -1,5 +0,0 @@

-checking

-

-1

-

-updating this one

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git diff --staged

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git add new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git diff --staged

**diff --git a/git\_clone\_repo/new\_file b/git\_clone\_repo/new\_file**

**index c3e9162..e69de29 100644**

**--- a/git\_clone\_repo/new\_file**

**+++ b/git\_clone\_repo/new\_file**

@@ -1,5 +0,0 @@

-checking

-

-1

-

-updating this one

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

Changes to be committed:

(use "git restore --staged <file>..." to unstage)

modified: new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git reset HEAD new\_file

Unstaged changes after reset:

M git\_clone\_repo/new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

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: new\_file

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git add new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git commit -m "commit-msg-3"

[new\_branch 45254f6] commit-msg-3

1 file changed, 5 deletions(-)

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git log

commit 45254f6a5b866b9b4c99636f6d131e5de15a0585 (HEAD -> new\_branch)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 13:05:05 2021 +0530

commit-msg-3

commit df5e56d1e815edd816d503563420e9d7dfeb44a4

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:33:50 2021 +0530

commit-msg-2

commit 6356d4422133f36ffb844e2d2ac8babb1c22ab59

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:30:53 2021 +0530

commit-msg-1

commit f5efc15236808aa995b2906a9d550489212d2886

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

0 [sig] bash 1205! sigpacket::process: Suppressing signal 18 to win32 process (pid 20628)

commit 45254f6a5b866b9b4c99636f6d131e5de15a0585 (HEAD -> new\_branch)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 13:05:05 2021 +0530

commit-msg-3

commit df5e56d1e815edd816d503563420e9d7dfeb44a4

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:33:50 2021 +0530

commit-msg-2

commit 6356d4422133f36ffb844e2d2ac8babb1c22ab59

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 12:30:53 2021 +0530

commit-msg-1

commit f5efc15236808aa995b2906a9d550489212d2886

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

Merge branch 'greeting'

commit 5d576d4a70262c3386ec3918a3e2a18f599a59b5

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:56:43 2021 +0530

grv2

commit f7d096a6d4105655231e7034cdadf323baacf6c4 (greeting)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:55:40 2021 +0530

grv1

commit 84c9c70fb7355cb4ac2e1bb95641389ee5fc4cd1 (grv)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:43:46 2021 +0530

grv

(END)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

Merge branch 'greeting'

commit 5d576d4a70262c3386ec3918a3e2a18f599a59b5

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:56:43 2021 +0530

grv2

commit f7d096a6d4105655231e7034cdadf323baacf6c4 (greeting)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:55:40 2021 +0530

grv1

commit 84c9c70fb7355cb4ac2e1bb95641389ee5fc4cd1 (grv)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:43:46 2021 +0530

grv

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Wed Nov 10 10:41:06 2021 +0530

commit-msg1

commit 260592c225a0d1add451c1fab6b31a38bafd9e83

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:03:58 2021 +0530

assigment1-2

commit f2e1a6c1847122cf593bc057fcb7695effccb771 (master)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 20:00:13 2021 +0530

assignment1-1

commit 1ff7315d09974bb0804ecc9a952a4d9671060b01

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 19:53:37 2021 +0530

assignment1

commit 4e7a3089c42dd728121eec81ff232c034bc2cdb9

Merge: 5d576d4 f7d096a

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:58:22 2021 +0530

Merge branch 'greeting'

commit 5d576d4a70262c3386ec3918a3e2a18f599a59b5

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:56:43 2021 +0530

grv2

commit f7d096a6d4105655231e7034cdadf323baacf6c4 (greeting)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:55:40 2021 +0530

grv1

commit 84c9c70fb7355cb4ac2e1bb95641389ee5fc4cd1 (grv)

Author: Gaurav verma <gaurav.verma@zemosolabs.com>

Date: Tue Nov 9 17:43:46 2021 +0530

grv

(END)

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

gi

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ echo 4>new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ cat new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

nothing to commit, working tree clean

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git checkout new\_file

Updated 0 paths from the index

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

nothing to commit, working tree clean

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$

KATA 3-TASK:

You again live in your own branch, this time we will be doing a bit of juggling with branches, to show how lightweight branches are in git. Hint: git switch will make you switch from one branch to another.

1. Use git branch to see the two branches that are relevant for this exercise
2. What branch are you on?
3. Use git branch mybranch to create a new branch called *mybranch*
4. Use git branch again to see the new branch created.
5. Use git switch mybranch to switch to your new branch.
6. How does the output from git status change when you switch between the *master* and the new branch that you have created?
7. How does the workspace change when you change between the two branches?
8. Make sure you are on your *mybranch* branch before you continue.
9. Create a file called file1.txt with your name.
10. Add the file and commit with this change.
11. Use git log --oneline --graph to see your branch pointing to the new commit.
12. Switch back to the branch called *master*.
13. Use git log --oneline --graph and notice how the commit you made on the *mybranch* branch is missing on the *master* branch.
14. Make a new file called file2.txt and commit that file.
15. Use git log --oneline --graph --all to see your branch pointing to the new commit, and that the two branches now have different commits on them.
16. Switch to your branch *mybranch*.
17. What happened to your working directory? Can you see your file2.txt?
18. Use git diff mybranch master to see the difference between the two branches.

KATA 3-SOLUTION:

Answering to 6th & 7th questions asked, since new branch created is totally a new branch so it’s git status will be empty as shown while git status in master will show respective changes. And we move ahead with the later qsn then clearly the workspace remains similar to other branches created apart from master.

Also asked in the couple of last questions, we do observe that creating a new file in master does not means that same will be created in other branches as well. Also branch commit are differently made in master than in other branches.

Rest of all questions are output based which I’ve pasted here:

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git branch

greeting

grv

master

\* new\_branch

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git status

On branch new\_branch

nothing to commit, working tree clean

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git branch mybranch

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git branch

greeting

grv

master

mybranch

\* new\_branch

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git switch mybranch

Switched to branch 'mybranch'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git status

On branch mybranch

nothing to commit, working tree clean

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git switch master

warning: unable to rmdir 'git\_clone\_repo/zemoso-training': Directory not empty

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

./

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ gti switch my

bash: gti: command not found

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch mybranch

Switched to branch 'mybranch'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ ls

new\_file zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git switch master

warning: unable to rmdir 'git\_clone\_repo/zemoso-training': Directory not empty

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ ls

zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch new\_branch

Switched to branch 'new\_branch'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ ls

new\_file zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (new\_branch)

$ git switch mybranch

Switched to branch 'mybranch'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ ls

new\_file zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ rm -rf new\_file

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ nano file1.txt

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git add file1.txt

warning: LF will be replaced by CRLF in git\_clone\_repo/file1.txt.

The file will have its original line endings in your working directory

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git commit -m "commit-msg-4"

[mybranch 481257a] commit-msg-4

1 file changed, 1 insertion(+)

create mode 100644 git\_clone\_repo/file1.txt

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git log --oneline --graph

\* 481257a (HEAD -> mybranch) commit-msg-4

\* 45254f6 (new\_branch) commit-msg-3

\* df5e56d commit-msg-2

\* 6356d44 commit-msg-1

\* f5efc15 commit-msg1

\* 260592c assigment1-2

\* f2e1a6c (master) assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a (greeting) grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 (grv) grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ git switch master

warning: unable to rmdir 'git\_clone\_repo/zemoso-training': Directory not empty

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --oneline --graph

\* f2e1a6c (HEAD -> master) assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a (greeting) grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 (grv) grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ nano file2.txt

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ ls

zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ touch file2.txt

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ ls

file2.txt zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git add file2.txt

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git commit -m "commit-msg-5"

[master fa47524] commit-msg-5

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 git\_clone\_repo/file2.txt

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --oneline --graph

\* fa47524 (HEAD -> master) commit-msg-5

\* f2e1a6c assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a (greeting) grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 (grv) grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch mybranch

Switched to branch 'mybranch'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$ ls

file1.txt new\_file zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (mybranch)

$

KATA 4-TASK:

1. Create a (feature)branch called feature/uppercase
2. Switch to this branch
3. What is the output of git status?
4. Edit the greeting.txt to contain an uppercase greeting
5. Add greeting.txt files to staging area and commit
6. What is the output of git branch?
7. What is the output of git log --oneline --graph --all

*Remember: you want to pull in the commit on the feature branch into master. The command 'git merge [branch name]' takes one branch as argument from which it takes commits. The commits are applied to the branch pointed to by HEAD (currently checked out branch).*

1. Switch to the master branch
2. Use cat to see the contents of the greetings
3. Diff the branches
4. Merge the branches
5. Use cat to see the contents of the greetings
6. Delete the uppercase branch

KATA 4-SOLUTION:

No as such theoretical questions here, So pasting here the further output based questions given above.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git branch uppercase

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch uppercase

Switched to branch 'uppercase'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git status

On branch uppercase

Untracked files:

(use "git add <file>..." to include in what will be committed)

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ nano greeting

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git add greeting

warning: LF will be replaced by CRLF in git\_clone\_repo/greeting.

The file will have its original line endings in your working directory

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git commit -m "commit-msg-5"

[uppercase d67e7f0] commit-msg-5

1 file changed, 1 insertion(+)

create mode 100644 git\_clone\_repo/greeting

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git branch

greeting

master

mybranch

\* uppercase

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git log --oneline --graph

\* d67e7f0 (HEAD -> uppercase) commit-msg-5

\* fa47524 (master) commit-msg-5

\* f2e1a6c assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a (greeting) grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git switch uppercase

Already on 'uppercase'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git switch master

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git pull master uppercase

fatal: 'master' does not appear to be a git repository

fatal: Could not read from remote repository.

Please make sure you have the correct access rights

and the repository exists.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git merge master uppercase

Updating fa47524..d67e7f0

Fast-forward

git\_clone\_repo/greeting | 1 +

1 file changed, 1 insertion(+)

create mode 100644 git\_clone\_repo/greeting

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ cat greeting

HEY!WELCOME

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git diff master uppercase

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git branch -D greeting

Deleted branch greeting (was f7d096a).

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git branch -D uppercase

Deleted branch uppercase (was d67e7f0).

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$

KATA 5-TASK:

You again live in your own branch, this time we will be doing a bit of juggling with branches, to show how lightweight branches are in git.

1. Create a branch called greeting and check it out
2. Edit the greeting.txt to contain your favorite greeting
3. Add greeting.txt files to the staging area
4. Commit
5. Switch back to the master branch
6. Create a file README.md with information about this repository
7. Add the README.md file to staging area and make the commit
8. What is the output of git log --oneline --graph --all?
9. Diff the branches
10. Merge the greeting branch into master
11. What is the output of git log --oneline --graph --all now? Observe the extra merge commit created with the message "Merge branch 'greeting'".

KATA 5-SOLUTION:

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git branch greeting

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git checkout greeting

Switched to branch 'greeting'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ ls

file2.txt greeting zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ nano greeting

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ git add greeting

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ git commit -m "commit-msg-6"

[greeting 241f304] commit-msg-6

1 file changed, 1 insertion(+), 1 deletion(-)

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ gti switch masster

bash: gti: command not found

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ git switch masster

fatal: invalid reference: masster

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (greeting)

$ git switch master

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ nano README.md

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git add README.md

warning: LF will be replaced by CRLF in git\_clone\_repo/README.md.

The file will have its original line endings in your working directory

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git commit -m "commit-msg-7"

[master 0cf0975] commit-msg-7

1 file changed, 1 insertion(+)

create mode 100644 git\_clone\_repo/README.md

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --oneline --graph

\* 0cf0975 (HEAD -> master) commit-msg-7

\* d67e7f0 commit-msg-5

\* fa47524 commit-msg-5

\* f2e1a6c assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git diff master greeting

fatal: ambiguous argument 'greeting': both revision and filename

Use '--' to separate paths from revisions, like this:

'git <command> [<revision>...] -- [<file>...]'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git merge master greeting

Merge made by the 'recursive' strategy.

git\_clone\_repo/greeting | 2 +-

1 file changed, 1 insertion(+), 1 deletion(-)

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --oneline --graph

\* 3df0665 (HEAD -> master) Merge branch 'greeting'

|\

| \* 241f304 (greeting) commit-msg-6

\* | 0cf0975 commit-msg-7

|/

\* d67e7f0 commit-msg-5

\* fa47524 commit-msg-5

\* f2e1a6c assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 grv

KATA 6,7-TASK:

These are based on merge conflicts including some specific files which were not found. Some baic commands which were involved with it are like: **git diff head~2 head**

KATA 8-TASK:

1. Which branches exist?
2. Look at the log for the master branch
3. Check out the uppercase branch
4. How does the log compare to the log on the master branch?
5. Rebase your uppercase branch with the master (git rebase master)
6. What did just happen? Draw it!
7. Now switch to the master branch
8. Merge uppercase into master
9. What does the log look like now?

KATA 8-SOLUTION:

So here in this kata we are basically trying to filter out the differncce between merge and rebase which simple means that in merge a new commit is created to bring the changes from one branch to other while using rebase those changes related commit are just re-written. Below is the attached practical implementation.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch uppercase

Switched to branch 'uppercase'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ touch test

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git add test

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git commit -m "commit-msg7"

[uppercase 941d74d] commit-msg7

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 git\_clone\_repo/test

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git switch master

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git rebase uppercase

Successfully rebased and updated refs/heads/master.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --oneline --graph

\* 941d74d (HEAD -> master, uppercase) commit-msg7

\* 3df0665 Merge branch 'greeting'

|\

| \* 241f304 (greeting) commit-msg-6

\* | 0cf0975 commit-msg-7

|/

\* d67e7f0 commit-msg-5

\* fa47524 commit-msg-5

\* f2e1a6c assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch uppercase

Switched to branch 'uppercase'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git touch test2

git: 'touch' is not a git command. See 'git --help'.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git add test2

fatal: pathspec 'test2' did not match any files

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ ls

README.md file2.txt greeting mc2 test zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git touch test2

git: 'touch' is not a git command. See 'git --help'.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ touch test2

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git add test2

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git commit -m "commit-msg-8"

[uppercase dca42df] commit-msg-8

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 git\_clone\_repo/test2

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git switch master

Switched to branch 'master'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git merge master uppercase

Updating 941d74d..dca42df

Fast-forward

git\_clone\_repo/test2 | 0

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 git\_clone\_repo/test2

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --oneline --graph

\* dca42df (HEAD -> master, uppercase) commit-msg-8

\* 941d74d commit-msg7

\* 3df0665 Merge branch 'greeting'

|\

| \* 241f304 (greeting) commit-msg-6

\* | 0cf0975 commit-msg-7

|/

\* d67e7f0 commit-msg-5

\* fa47524 commit-msg-5

\* f2e1a6c assignment1-1

\* 1ff7315 assignment1

\* 4e7a308 Merge branch 'greeting'

|\

| \* f7d096a grv1

\* | 5d576d4 grv2

|/

\* 84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$

KATA 9-TASK:

1. Use git log --decorate --oneline to look at the history
2. Use cat to view the content of greeting.txt
3. Use git revert on the newest commit, to remove the changes the last commit added
4. Use git log --decorate --oneline to view the history
5. Did the revert command add or remove a commit?
6. Use cat to view the content of greeting.txt
7. Use ls to see the content of the workspace
8. Use git log --decorate --oneline to find the sha of the commit adding credentials to the repository
9. Use git revert to revert the commit that added the credentials
10. Use git log --decorate --oneline to view the history
11. Use ls to see the content of the workspace
12. How many commits were added or changed by the last revert?
13. Use git show with the sha of the commit you reverted to see that the credentials file is stilll in the history
14. As you have now reverted the credentials file, so it is removed from your working directory, is it also removed from git?

KATA 9-SOLUTION:

Answering to the 5th question, the revert added a new commit point as per logs

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --decorate --oneline

dca42df (HEAD -> master, uppercase) commit-msg-8

941d74d commit-msg7

3df0665 Merge branch 'greeting'

0cf0975 commit-msg-7

241f304 (greeting) commit-msg-6

d67e7f0 commit-msg-5

fa47524 commit-msg-5

f2e1a6c assignment1-1

1ff7315 assignment1

4e7a308 Merge branch 'greeting'

5d576d4 grv2

f7d096a grv1

84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ cat greeting

EDITED-HEY!WELCOME

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git revert

usage: git revert [<options>] <commit-ish>...

or: git revert <subcommand>

--quit end revert or cherry-pick sequence

--continue resume revert or cherry-pick sequence

--abort cancel revert or cherry-pick sequence

--skip skip current commit and continue

--cleanup <mode> how to strip spaces and #comments from message

-n, --no-commit don't automatically commit

-e, --edit edit the commit message

-s, --signoff add a Signed-off-by trailer

-m, --mainline <parent-number>

select mainline parent

--rerere-autoupdate update the index with reused conflict resolution if possible

--strategy <strategy>

merge strategy

-X, --strategy-option <option>

option for merge strategy

-S, --gpg-sign[=<key-id>]

GPG sign commit

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git revert dca42df

Removing git\_clone\_repo/test2

[master 3cf9aed] Revert "commit-msg-8"

1 file changed, 0 insertions(+), 0 deletions(-)

delete mode 100644 git\_clone\_repo/test2

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git log --decorate --oneline

3cf9aed (HEAD -> master) Revert "commit-msg-8"

dca42df (uppercase) commit-msg-8

941d74d commit-msg7

3df0665 Merge branch 'greeting'

0cf0975 commit-msg-7

241f304 (greeting) commit-msg-6

d67e7f0 commit-msg-5

fa47524 commit-msg-5

f2e1a6c assignment1-1

1ff7315 assignment1

4e7a308 Merge branch 'greeting'

5d576d4 grv2

f7d096a grv1

84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ cat test2

cat: test2: No such file or directory

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git switch uppercase

Switched to branch 'uppercase'

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ cat test2

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$

KATA 10,11,12-TASK(RESET,CLEANING,AMMEND):

🡪Here cleaning can be done using CLEAN in order to clean some unrequired access generated files.

Some of it’s commands are:

🡪git clean -n

🡪git clean -n -d

🡪git clean

While CLEAR can be used to clear the screen as well.

RESET:

Git reset –hard

Git reset –soft

Git reset –mixed

AMMEND :

If a commit has been made already and there is a need to add some more changes to already made commit. Then we can go with: git commit amend

KATA 13,14,15-(SQUASHING,STASHING)

The interactive rebase gives **you a script that it's going to run**. It will start at the commit you specify on the command line ( HEAD~3 ) and replay the changes introduced in each of these commits from top to bottom.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git log --decorate --oneline

dca42df (HEAD -> uppercase) commit-msg-8

941d74d commit-msg7

3df0665 Merge branch 'greeting'

0cf0975 commit-msg-7

241f304 (greeting) commit-msg-6

d67e7f0 commit-msg-5

fa47524 commit-msg-5

f2e1a6c assignment1-1

1ff7315 assignment1

4e7a308 Merge branch 'greeting'

5d576d4 grv2

f7d096a grv1

84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git rebase -i HEAD~4

Successfully rebased and updated refs/heads/uppercase.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git status

On branch uppercase

Untracked files:

(use "git add <file>..." to include in what will be committed)

mc2

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git rebase -i HEAD~4

[detached HEAD 610f997] commit-msg-6

Date: Wed Nov 10 14:27:59 2021 +0530

3 files changed, 1 insertion(+), 1 deletion(-)

create mode 100644 git\_clone\_repo/test

create mode 100644 git\_clone\_repo/test2

Successfully rebased and updated refs/heads/uppercase.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git status

On branch uppercase

Untracked files:

(use "git add <file>..." to include in what will be committed)

mc2

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$ git log --decorate --oneline

610f997 (HEAD -> uppercase) commit-msg-6

0cf0975 commit-msg-7

d67e7f0 commit-msg-5

fa47524 commit-msg-5

f2e1a6c assignment1-1

1ff7315 assignment1

4e7a308 Merge branch 'greeting'

5d576d4 grv2

f7d096a grv1

84c9c70 grv

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (uppercase)

$

STASHING:

Stashing takes away our non-committed work to a different area which we’ve the privilege to re-apply later on.

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ ls

README.md file2.txt greeting mc2 test zemoso-training/

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

mc2

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ nano test

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (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: test

Untracked files:

(use "git add <file>..." to include in what will be committed)

mc2

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git stash push

warning: LF will be replaced by CRLF in git\_clone\_repo/test.

The file will have its original line endings in your working directory

Saved working directory and index state WIP on master: 3cf9aed Revert "commit-msg-8"

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git status

On branch master

Untracked files:

(use "git add <file>..." to include in what will be committed)

mc2

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git stash list

stash@{0}: WIP on master: 3cf9aed Revert "commit-msg-8"

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$ git stash apply

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: test

Untracked files:

(use "git add <file>..." to include in what will be committed)

mc2

zemoso-training/

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

gv231@LAPTOP-M3ET0I21 MINGW64 ~/git\_demo/git\_clone\_repo (master)

$