**1. Git checkout to get the original state back :**

Create a branch subtract-feature

git branch subtract-feature

Made change to the subtract function being in master branch only and say you made the bad changes

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

def add(x,y):

pass

def subtract(x,y):

abcdefff

ghjkkkkkkss

def multiply(x,y):

pass

def divide(x,y):

pass

def square(x,y):

pass

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

git status

git diff ( shows the changes made)

git checkout calc.py

git status

git diff (no diffs means the file went to original state)

**2. Changing the wrong committed messages :**

Make a valid change to the subtract function

return x - y

*git status*

*git add -A*

*git commit -m “Completed multiply function” (A bad commit with wrong message)*

*git commit --amend -m “Completed subtract function” (using –amend to change the message)*

*git log*

**3. Git Cherry Picking**

Since we made changes of subtract in master but it was meant to be done on subtract feature. This is very common in day to day work

*git log ( we have to copy the first 6-7 character from the hash)*

*git checkout subtract-feature*

*git log*

*git cherry-pick 1b818d3b (first 7 char. Of hash from the last commit on the master using git log)*

*git log*

**4. Git Reset**

Git reset are of three types : soft, mixed(default) and hard

*Let’s get back to master and reset it back to the state were we have not made the subtract feature changes in master*

*git log (and pick the hash of the commit older than the “completed subtract feature one”*

*git reset --soft 2e7520782 ( will take to stage where we see the new file if added and file in modified state)*

*git status*

*git reset 2e7520782 ( it’s a mixed reset)*

*git status*

*git reset --hard 2e7520782 ( be careful when using it, it will remove all the tracked file changes before commit for the passed hash id)*

*git status*

**5. Getting rid of untracked files**

*git clean -df*  (will remove the untracked file)

git status

**6. Recovering from hard reset**

*git reflog* (add All file to staging area)

*git checkout <hash id> (will detatch the head till the hash id commit)*

*git branch backup (to create a backup branch from the head)*

*git branch (to see the branch created and currently its on detached head)*

*git checkout master ( return to master from detached head after taking the backup)*

*git checkout backup (go to branch backup)*

*git log*

**7. Git Revert**

*git reflog*

*git revert <hash-id> (the commit that you want to revert to without changing history)*

*git log (you will see a new revert hash created and old hashes are intact)*

git diff <old-hash> <new-hash>

**Working with Stash**

Lets say in working directory in master we have a calc.py that contains

----------------------------

def add(x,y):

pass

def subtract(x,y):

pass

def multiply(x,y):

pass

def divide(x,y):

pass

def square(x,y):

pass

-----------------------------

*git branch add*

*git checkout add*

*make changes to add function : inplace of* ***pass*** *write* ***return x+y***

*git diff*

*git stash save “worked on add function”*

*git diff ( no changes will show up but for good)*

*git status*

*git stash list ( will show stash@{id}) you can switch branches work and come back*

*git stash apply stash@{id} (take you back to the changes)*

*git stash list ( will show the id again)*

*git checkout -- . ( going back to old change)*

*git stash list*

*git stash pop ( apply and drop the stash)*

*git stash list*

*git diff*

*Lets make change to rest of function*

----------------------------

def add(x,y):

return x + y

def subtract(x,y):

return x - y

def multiply(x,y):

return x \* y

def divide(x,y):

return x / y

*git stash save “Calc Functions”*

*git stash list*

*now lets add the square functions that we have not added*

def add(x,y):

pass

def subtract(x,y):

pass

def multiply(x,y):

pass

def divide(x,y):

pass

def square(x,y):

pass

*git diff*

*git stash save “added square function”*

*git stash list (you can see the calc function is pushe down to id 1 from 0 and square function takes id 0)*

*git stash drop stash@{id} (put the id that you want to drop, say we want to drop square fucnction change is that is 0)*

*git stash list*

*git stash clear ( will remove all the stashes. Be careful when running this, ensure if you want to get rid of all the stash changes)*

*git stash list*

Git\_FixingCommonMistakesAndUndoingBadCommits