Git Notes

Git branch

* **Branch --vv** shows each local branch and the remote it is connected to
* **Branch --set-upstream-to=origin/remote-branch-name**, changes what remote branch the local branch is pointing to

Git Push

* **push**
* **Push origin HEAD: remote-branch-name,** pushes the branch to remote if the remote branch has a different name

Git Pull

* **Pull** pulls form origin to local

Git Commit

* Commit -m “…” commits current staged changed with the title
* Commit -am “…” add and commits all changes with title
* Commit –amend -m “…” if it is only local branch, you can change the commit name

Git Merge

* Merge 2nd\_branch\_name merges in with the current branch
* Merge --abort aborts the merge and reverts the changes
* Merge -- quite quits the merge but saves the current changes
* Merge -- continue continues to the next issue

Git Add

* add . adds all of the changes
* add file\_name adds a specific file

Git Checkout

* checkout “”
* ***checkout --patch <source\_branch> <path/to/file>***
* checkout -b “…” creates new branch if one not found

Git Revert

* ***revert commit-id***  *reverts the intended commit*
* ***git revert starting-commit-hash..ending-commit-hash*** *reverts a range*
* ***--no-edit*** *reverts the commit instantly*
* ***--abort***

Git Reset

* **Reset –hard commit\_name** moves head to the inputted commit and deletes the ones after it
* ***reset --hard HEAD~N*** *replacing N with the number of commits you want to remove*
* **reset --soft HEAD~N** This command moves the branch pointer back, but keeps the changes introduced by the undone commits

Git Stash

* only for local changed uncommitted
* stash temporary saves changes to files allowing you to leave the branch without discarding needed data
* stash pop reprints the stashed data back onto branch/deletes form list
* stash apply reprints the stashed info without removing it from the list
* stash save “” saves the information to a name
* stash list shows all of the saved names
* stash drop (stash\_name) deletes names from the stash list

Git Fetch

* fetch origin similar to pull but saves the changes to a clipboard instead of directly inserting it into the branch, use a git merge to combine the data

Git Bisect

* used to go through each commit and determine where a bug originated from
* bisect start start the bisect
* git bisect bad [commit]: enter a commit that has the bug(most current)
* git bisect good [commit]: enter a commit that doesn’t have the bug(earlier)
* git bisect reset clear the bisect and stop
* git bisect skip skip the current commit
* ***git bisect run <command>*** *automates the process by running a command on each commit and testing it*

Git diff

* git diff master..local\_branch shows the difference between master and branch

Git log

* log shows all commit and repository history
* log –oneline shows only the commits in order