Configuration: git config

--global user.name “blahblahblah” 🡪 set user name

--global user.email “blahblahblah” 🡪 set user email

--global color.ui true 🡪 activate colors in terminal

--global core.editor blahblahblah 🡪 set preferred text editor

--list 🡪 check all your settings

git help 🡪 overview of git commands

git init 🡪 new project

git status 🡪 see changes

git add [FILENAME] 🡪 select files for snapshot

git add --all 🡪 all files for snapshot

git commit -m “blahblahblah” 🡪 create snapshot, -m is the message

git diff 🡪 see all changes made since the last commit

cat .gitignore 🡪 create a file including what names/types of files you would like to ignore

\*Put it in the main directory of the repository

put “git” in front of commands to move stuff around so git can keep track

git rm filename.txt

git rm \*.txt

git mv oldfilename.txt newfilename.txt

git commit --amend 🡪 fix a small error on a commit

git reset HEAD [filename] 🡪 “unstage a file”

git checkout 🡪 fetches a fresh copy from the repository, gets rid of recent edits

git checkout [name of branch] 🡪 switches which branch you’re working on

git checkout [ID of previous commit] 🡪 check out a previous version of the project

git clone 🡪 download a local copy of a repository

git pull 🡪 get the latest version of the repository

git push 🡪 let everyone see your latest commit

git stash 🡪 stash incomplete changes away

git stash apply 🡪 apply your stashed changes to the repository

create\_repository.sh 🡪 located in CSB/git/data and creates a repository

git branch [name of branch] 🡪 create new branch

git branch 🡪 shows a list of branches (\*marks the one you are working on)

git branch -d [name of branch] 🡪 delete branch you no longer need

git -d [name of branch] 🡪 may also delete it, unclear

git log --oneline --decorate 🡪 see status/history of your branch

git log --oneline --decorate --all --graph 🡪 graph summarizing the state of all branches

git merge [name of branch] 🡪 be in master to use this an merge a branch back with master

CONFLICTS: open file in text editor, removing markers, resolve, stage, and commit

<<<<<<<< beginning of area that needs to be fixed

version one of the line that failed to merge

========

version two of the line that failed to merge

>>>>>>>> end of the area that needs to be fixed