

# Michael Deitzel

110075

## LAB 1

### Lab 1 Solutions

#### 1. git init

```
C:\Users\Michael\Desktop\lab1>git init
Initialized empty Git repository in C:/Users/Michael/Desktop/lab1/.git/
```

git init generates .git hidden folder

#### 2. git status

```
C:\Users\Michael\Desktop\lab1>git status
On branch master
```

No commits yet

nothing to commit (create/copy files and use "git add" to track)

git status looks at changes on working directory

#### 3. git diff

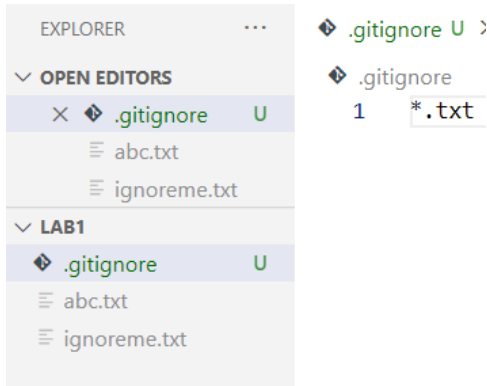
```
C:\Users\Michael\Desktop\lab1>git log --oneline
f9d1990 (HEAD -> master) amended gitignore init commit
```

```
C:\Users\Michael\Desktop\lab1>git diff .gitignore
diff --git a/.gitignore b/.gitignore
index 314f02b..b92c542 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1,2 @@
-*.txt
\ No newline at end of file
+*.txt
+java
\ No newline at end of file
```

git diff add a file to see the difference by creating java file and adding to .gitignore

java treats \*.txt as brand new as well. usually don't use difficult to know what changes are made

#### 4. Create .gitignore file, and notice how it works



.gitignore file – put files in here that don't want to go into the repository

#### 5. git add

```
C:\Users\Michael\Desktop\lab1>git add .gitignore
```

```
C:\Users\Michael\Desktop\lab1>git status
On branch master
```

```
No commits yet
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitignore
```

git add – adds file from working area to staging area

#### 6. git commit

```
C:\Users\Michael\Desktop\lab1>git commit -m "added gitignore file"
[master (root-commit) 05278d3] added gitignore file
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```

git commit – adds file from staging area to repository

#### 7. git commit -amend (updating the last commit)

```
C:\Users\Michael\Desktop\lab1>git commit --amend -m "amended gitignore init commit"
[master f9d1990] amended gitignore init commit
Date: Mon Oct 25 21:41:08 2021 -0500
 1 file changed, 1 insertion(+)
 create mode 100644 .gitignore
```

git commit amend – update last commit -m allows to change the message

## 8. git log

```
C:\Users\Michael\Desktop\lab1>git log
commit 05278d38b9a0b078ca41c8a468f990608789effc (HEAD -> master)
Author: mjdeitz <mjdeitz@yahoo.com>
Date: Mon Oct 25 21:41:08 2021 -0500
```

added gitignore file

git log – show all changes made to local repository

## 9. git tag

```
PS C:\Users\Michael\Desktop\lab1> git tag -a firstcommit -m "first" 4f6fe01
PS C:\Users\Michael\Desktop\lab1> git log --oneline
6e8e001 (HEAD -> master) console.log 3
f5aeac6 hello.js init commit
4f6fe01 (tag: firstcommit) hello.js init commit
56b6413 ignore java folder
f9d1990 amended gitignore init commit
```

git tag – gives an id to a change / easier to use versus hash code

## 10. git branch

```
C:\Users\Michael\Desktop\lab1>git branch
* master
  shoppingcart
  user-management
```

```
C:\Users\Michael\Desktop\lab1>git branch new-branch
```

## 11. git checkout

```
C:\Users\Michael\Desktop\lab1>git checkout master
Switched to branch 'master'
```

## 12. git merge – no conflict

```
PS C:\Users\Michael\Desktop\lab1> git merge shoppingcart
Merge made by the 'recursive' strategy.
 shoppingcart.js | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 shoppingcart.js
```

## i. Create a conflict and solve the conflict

```
C:\Users\Michael\Desktop\lab1>git checkout master
Switched to branch 'master'
```

```
C:\Users\Michael\Desktop\lab1>git merge shoppingcart
Auto-merging shoppingcart.js
CONFLICT (content): Merge conflict in shoppingcart.js
Automatic merge failed; fix conflicts and then commit the result.
```

```
C:\Users\Michael\Desktop\lab1>git add .
```

```
C:\Users\Michael\Desktop\lab1>git commit -m "merge changes from shoppingcart"
[master 9e1afe0] merge changes from shoppingcart
```

```
C:\Users\Michael\Desktop\lab1>git merge shoppingcart
Already up to date.
```

## 13. Undo changes from a certain commit git revert

```
C:\Users\Michael\Desktop\lab1>git log --oneline
0725e2a (HEAD -> master) git revert test again
8a05a63 Revert "prepare for revert test"
9cea199 prepare for revert test
9e1afe0 merge changes from shoppingcart
491f270 (shoppingcart) shopping cart changes
e096c1a changes on master
29079a4 (new-branch) Merge branch 'shoppingcart'
9da9d5c shoppingcart init commit
6e8e001 (user-management) console.log 3
f5aeac6 hello.js init commit
4f6fe01 (tag: firstcommit) hello.js init commit
56b6413 ignore java folder
f9d1990 amended gitignore init commit
```

```
C:\Users\Michael\Desktop\lab1>git revert 0725e2a
[master e449dba] Revert "git revert test again change"
1 file changed, 1 insertion(+), 2 deletions(-)
```

```
C:\Users\Michael\Desktop\lab1>git log --oneline
e449dba (HEAD -> master) Revert "git revert test again change"
0725e2a git revert test again
8a05a63 Revert "prepare for revert test"
9cea199 prepare for revert test
9e1afe0 merge changes from shoppingcart
491f270 (shoppingcart) shopping cart changes
e096c1a changes on master
29079a4 (new-branch) Merge branch 'shoppingcart'
9da9d5c shoppingcart init commit
6e8e001 (user-management) console.log 3
f5aeac6 hello.js init commit
4f6fe01 (tag: firstcommit) hello.js init commit
56b6413 ignore java folder
f9d1990 amended gitignore init commit
```

## 14. Erase commits with git reset

```
C:\Users\Michael\Desktop\lab1>git log --oneline
e449dba (HEAD -> master) Revert "git revert test again change"
0725e2a git revert test again
8a05a63 Revert "prepare for revert test"
9cea199 prepare for revert test
9e1afe0 merge changes from shoppingcart
491f270 (shoppingcart) shopping cart changes
e096c1a changes on master
29079a4 (new-branch) Merge branch 'shoppingcart'
9da9d5c shoppingcart init commit
6e8e001 (user-management) console.log 3
f5aeac6 hello.js init commit
4f6fe01 (tag: firstcommit) hello.js init commit
56b6413 ignore java folder
f9d1990 amended gitignore init commit
```

```
C:\Users\Michael\Desktop\lab1>git reset 9e1afe0
```

```
C:\Users\Michael\Desktop\lab1>git log --oneline
9e1afe0 (HEAD -> master) merge changes from shoppingcart
491f270 (shoppingcart) shopping cart changes
e096c1a changes on master
29079a4 (new-branch) Merge branch 'shoppingcart'
9da9d5c shoppingcart init commit
6e8e001 (user-management) console.log 3
f5aeac6 hello.js init commit
4f6fe01 (tag: firstcommit) hello.js init commit
56b6413 ignore java folder
f9d1990 amended gitignore init commit
```

## Practice with all the remote repository commands from the slides:

1. Create a remote repository

go to github account and create new repository. can have more than one.

2. git clone

use git clone to make cs445-lab-solns to be local / click on green code button and copy the github link / clones it from git hub to local folder on desktop

3. git remote

```
C:\Users\Michael\Desktop\cs445-lab-solns>git remote
origin
```

side note: on git hub the master branch name is NOT master it's called "main"

```
C:\Users\Michael\Desktop\cs445-lab-solns>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

#### 4. git push

I created hello.js file in local copy of cs445-lab-solns but when I refresh the github cs445-lab-solns it does not show. I need push to send it to github. /\*git push origin main\*/

next made changes to hello.js file added a second line then only need to use /\*git push\*/ to update github.

#### 5. git fetch

just sees changes on github repository - made changes to README.md by adding 1.Lab 1 - does not show on local README.md

```
C:\Users\Michael\Desktop\cs445-lab-solns>git fetch
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.38 KiB | 78.00 KiB/s, done.
From https://github.com/mjdeitz/cs445-lab-solns
   f48f27b..d344d49  main       -> origin/main
```

here I use git fetch and it shows changes but still no update on local README.md / need to use git merge then I see the changes that I made to the README.md file on github. so, git fetch only gets the changes but it does not add to local file must use git merge to do this

```
C:\Users\Michael\Desktop\cs445-lab-solns>git merge
Updating f48f27b..d344d49
Fast-forward
 README.md | 4 +++-
 1 file changed, 3 insertions(+), 1 deletion(-)
```

#### 6. git pull - this will do git fetch + git merge - usually use git pull

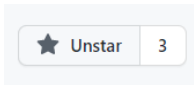
```
C:\Users\Michael\Desktop\cs445-lab-solns>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 725 bytes | 40.00 KiB/s, done.
From https://github.com/mjdeitz/cs445-lab-solns
   d344d49..3b531b6  main       -> origin/main
Updating d344d49..3b531b6
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
```

In summary, the ones used most are git clone, git push, git pull. git clone will take github repository and save it to local, git push will push changes to github, git pull will pull changes from github to local.

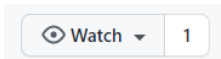
# Practice with all the third party repositories commands from the slides:

Assume we are working with the following upstream: <https://github.com/bellaxing/cs445-labs>

1. Star the repository

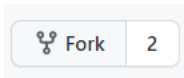


2. Watch the repository for any changes



3. Fork the repository

fork means copy bellas repository to my own account. Fork can NOT be done locally. only done from github. if bella makes changes after it will not show I would need to update



4. Clone your forked repository into your local machine

```
C:\Users\Michael\Desktop>git clone https://github.com/mjdeitz/cs445-labs.git
Cloning into 'cs445-labs'...
remote: Enumerating objects: 188, done.
remote: Counting objects: 100% (188/188), done.
remote: Compressing objects: 100% (116/116), done.
Receiving objects: 56% (106/188) | 115 (delta 30), pack-reused 0 | 45% (85/188)
Receiving objects: 100% (188/188), 95.73 KiB | 233.00 KiB/s, done.
Resolving deltas: 100% (72/72), done.
```

5. Create a new topic branch using studentId-studentName format, For example: 610001-JohnSmith

```
C:\Users\Michael\Desktop\cs445-labs>git branch 110075-MichaelDeitzel
```

6. Checkout your new branch

```
C:\Users\Michael\Desktop\cs445-labs>git checkout 110075-MichaelDeitzel
Switched to branch '110075-MichaelDeitzel'
```

7. Add a new file with your name, example: hello.js

```
C:\Users\Michael\Desktop\cs445-labs>git add hello.js
```

## 8. Commit the changes

```
C:\Users\Michael\Desktop\cs445-labs>git commit -m "changes in 110075-MichaelDeitzel branch"
[110075-MichaelDeitzel 646d0bf] changes in 110075-MichaelDeitzel branch
1 file changed, 1 insertion(+)
create mode 100644 hello.js
```

## 9. Merge your studentId-studentName branch with the main branch

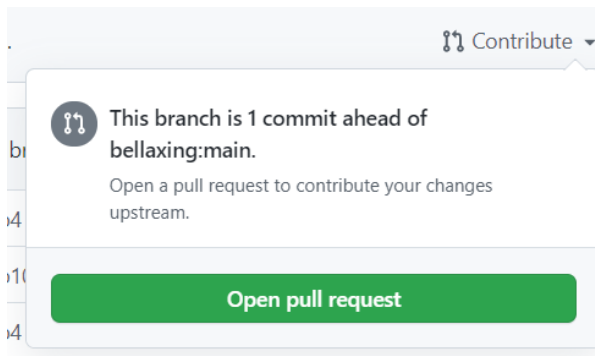
```
C:\Users\Michael\Desktop\cs445-labs>git merge 110075-MichaelDeitzel
Updating fc075c5..646d0bf
Fast-forward
 hello.js | 1 +
1 file changed, 1 insertion(+)
create mode 100644 hello.js
```

## 10. Push your changes to the remote repo

```
C:\Users\Michael\Desktop\cs445-labs>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes | 314.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/mjdeitz/cs445-labs.git
fc075c5..646d0bf main -> main
```

sends updated hello.js file to github cs445-labs

## 11. Create a new Pull Request to the upstream repo



create a new pull request can only be done on github!!! this request goes to bella. bella will approve and then merge it to bellas github repository.