

Practice with all the local repository commands from the slides:

1. git init

```
PS D:\MIU\CS445\CS445> git init
```

Reinitialized existing Git repository in D:/MIU/CS445/CS445/.git/

2. git status


```
PS D:\MIU\CS445\CS445> git status
```

```
PS D:\MIU\CS445\CS445> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

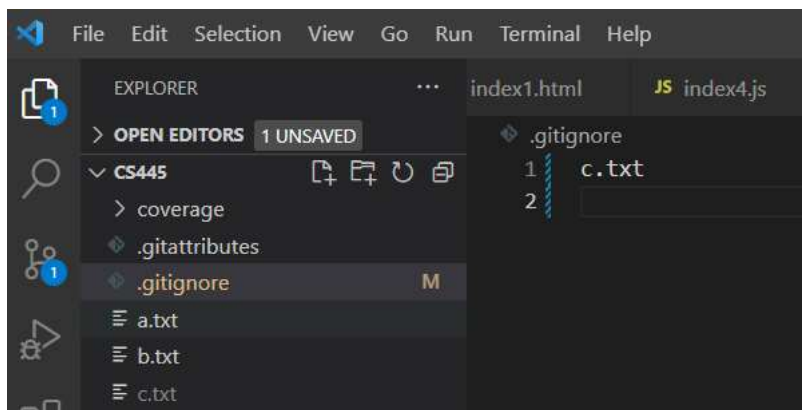
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.txt
        index1.html
        index2.html

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

3. git diff

	<pre>diff PS D:\MIU\CS445\CS445> git diff diff --git a/.gitignore b/.gitignore index e69de29..40816a2 100644 --- a/.gitignore +++ b/.gitignore @@ -0,0 +1 @@ +Hi \ No newline at end of file diff --git a/hello.txt b/hello.txt index e69de29..05db97c 100644 --- a/hello.txt +++ b/hello.txt @@ -0,0 +1,2 @@ +Hi +How are you diff --git a/index1.html b/index1.html index e69de29..d2d2991 100644 --- a/index1.html +++ b/index1.html @@ -0,0 +1,4 @@ +<!DOCTYPE html> +<html lang="en"> + +</html> \ No newline at end of file diff --git a/index2.html b/index2.html index e69de29..0e76edd 100644 --- a/index2.html +++ b/index2.html @@ -0,0 +1 @@ :...skipping... diff --git a/.gitignore b/.gitignore index e69de29..40816a2 100644 --- a/.gitignore +++ b/.gitignore @@ -0,0 +1 @@</pre>
---	--

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



5. git add

```
PS D:\MIU\CS445\CS445> git add .
PS D:\MIU\CS445\CS445> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   hello.txt
    new file:   index1.html
    new file:   index2.html

PS D:\MIU\CS445\CS445> git restore --staged .
PS D:\MIU\CS445\CS445> git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello.txt
    index1.html
    index2.html

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

6. git commit

```
PS D:\MIU\CS445\CS445> git commit -m "init commit"
[main 20b306e] init commit
3 files changed, 9 insertions(+)
create mode 100644 hello.txt
create mode 100644 index1.html
create mode 100644 index2.html
```

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

```
PS D:\MIU\CS445\CS445> git commit --amend -m "amended gitignore init"
[main fbadd55] amended gitignore init
Date: Tue Apr 26 14:01:20 2022 -0500
2 files changed, 4 insertions(+)
```

8. git log

```

PS D:\MIU\CS445\CS445> git log
commit fbadd55b37d3e6b3b8509fa9a63d287d94b6e6a2 (HEAD -> main)
Author: tam <ngocnguyen@miu.edu>
Date: Tue Apr 26 14:01:20 2022 -0500

    amended gitignore init

commit 4021c561da848750d1bae18daf78c48d8dfed424
Author: tam <ngocnguyen@miu.edu>
Date: Tue Apr 26 13:55:55 2022 -0500

    amended gitignore init

commit 789c8cdd85309fb22352208e7c7465184c217014
Author: tam <ngocnguyen@miu.edu>
Date: Tue Apr 26 13:53:05 2022 -0500

    init gitignore

commit 18409a98ea185b2fec1e4e61353739e0194b3798
Author: tam <ngocnguyen@miu.edu>
Date: Tue Apr 26 13:51:55 2022 -0500

    init gitignore

commit cf2b17d6a68315cac3d46fe957baf3834012b6c3
Author: tam <ngocnguyen@miu.edu>
Date: Tue Apr 26 09:43:36 2022 -0500

```

9. git tag

```

PS D:\MIU\CS445\CS445> git tag -a firstcommit -m "first" 4021c56
PS D:\MIU\CS445\CS445> git log --oneline
fbadd55 (HEAD -> main) amended gitignore init
4021c56 (tag: firstcommit) amended gitignore init
789c8cd init gitignore
18409a9 init gitignore
cf2b17d index1
9a1b014 (tag: index4Tag) index4.js
1c64e9b index 1
0757367 (origin/main) init commit
20b306e init commit
f6d8eee added gitignore file
ff4f180 Initial commit

```

10. git branch

```

PS D:\MIU\CS445\CS445> git branch
* main
PS D:\MIU\CS445\CS445> git branch rightBranch
PS D:\MIU\CS445\CS445> git branch leftBranch
PS D:\MIU\CS445\CS445> git branch
    leftBranch
* main
    rightBranch

```

11. git checkout

```

PS D:\MIU\CS445\CS445> git branch
leftBranch
* main
rightBranch
PS D:\MIU\CS445\CS445> git checkout leftBranch
Switched to branch 'leftBranch'
M .gitignore
PS D:\MIU\CS445\CS445> git branch
* leftBranch
main
rightBranch

```

12. git merge

```

PS D:\MIU\CS445\CS445> git switch leftBranch
Switched to branch 'leftBranch'
PS D:\MIU\CS445\CS445> git switch main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 7 commits.
(use "git push" to publish your local commits)
PS D:\MIU\CS445\CS445> git merge leftBranch
Updating fbadd55..f4abddd
Fast-forward
 .gitignore      | 2 --
 c.txt           | 0
 createdOnLeft-1.js | 0
 createdOnLeft.js | 0
 createdOnRight/fileOnRight.js | 0
 5 files changed, 2 deletions(-)
 create mode 100644 c.txt
 create mode 100644 createdOnLeft-1.js
 create mode 100644 createdOnLeft.js
 create mode 100644 createdOnRight/fileOnRight.js

```

12.1 merge conflict

```

JS index4.js  JS index5.js 3,! X  hello.txt  index2.html  index3.html
JS index5.js
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
1 | <<<<<<< HEAD (Current Change)
2 | console.log("modified on right")
3 | =====
4 | //(-----modified on leftBranch-----");
5 |
OUTPUT  DEBUG CONSOLE  PROBLEMS 3  TERMINAL
PS D:\MIU\CS445\CS445> git add .
PS D:\MIU\CS445\CS445> git add .
PS D:\MIU\CS445\CS445> git commit -m "init on main"
[main 64390d8] init on main
1 file changed, 1 insertion(+)
PS D:\MIU\CS445\CS445> git merge rightBranch
Auto-merging index5.js
CONFLICT (content): Merge conflict in index5.js
Automatic merge failed; fix conflicts and then commit the result.

```

12.2 solve the conflict


```
JS index4.js JS index5.js X hello.txt index2.htm
JS index5.js
1 console.log("modified on right");
2
3 //(-----modified on leftBranch-----");
4
5 console.log("modified on main");
6
7 console.log("modified on right");
8
9 console.log("modified on rightBranch ");
10
OUTPUT DEBUG CONSOLE PROBLEMS TERMINAL
PS D:\MIU\CS445\CS445> git add index5.js
PS D:\MIU\CS445\CS445> git commit -m "init index5.js again"
[main 6c56dc3] init index5.js again
```

```
PS D:\MIU\CS445\CS445> git merge rightBranch
Already up to date.
```

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

```
PS D:\MIU\CS445\CS445> git log --oneline
92059e1 (HEAD -> main) Revert "init index6 on main"
49ad1a3 init on main
6c56dc3 init index5.js again
64390d8 init on main
a5ce33f (rightBranch) init on rightBranch
afc6fe3 init on right
f260b21 init on main
f4abddd (leftBranch) init on leftBranch
fbdd55 amended gitignore init
4021c56 (tag: firstcommit) amended gitignore init
789c8cd init gitignore
18409a9 init gitignore
cf2b17d index1
9a1b014 (tag: index4Tag) index4.js
1c64e9b index 1
0757367 (origin/main) init commit
20b306e init commit
f6d8eee added gitignore file
ff4f180 Initial commit
PS D:\MIU\CS445\CS445> git revert HEAD
[main b9ba4bd] Revert "Revert "init index6 on main""
1 file changed, 1 insertion(+)
create mode 100644 index6.js
```

14. Erase commits with `git reset`

```

PS D:\MIU\CS445\CS445> git reset a5ce33f
Unstaged changes after reset:
M      .gitignore
M      index5.js
PS D:\MIU\CS445\CS445> git log --oneline
a5ce33f (HEAD -> main, rightBranch) init on rightBranch
afc6fe3 init on right
fbedd55 amended gitignore init
4021c56 (tag: firstcommit) amended gitignore init
789c8cd init gitignore
18409a9 init gitignore
cf2b17d index1
9a1b014 (tag: index4Tag) index4.js
1c64e9b index 1
0757367 (origin/main) init commit
20b306e init commit
f6d8eee added gitignore file
ff4f180 Initial commit

```

Practice with all the remote repository commands from the slides:

1. Create a remote repository

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *

Repository name *

 ngoctamnguyen ▾ / CS445-lab ✓

Great repository names are short and memorable. Need inspiration? How about [miniature-dollop](#)?

Description (optional)

☒  Public

Anyone on the internet can see this repository. You choose who can commit.

2. git clone

```
PS D:\MIU\CS445\CS445> git clone https://github.com/ngoctamnguyen/CS445-lab1
Cloning into 'CS445-lab1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
PS D:\MIU\CS445\CS445>
```

3. git remote

```
PS D:\MIU\CS445\CS445\cs445-lab1> git remote
origin
```

4. git push

```
PS D:\MIU\CS445\CS445> cd cs445-lab1
PS D:\MIU\CS445\CS445\cs445-lab1> ls

Directory: D:\MIU\CS445\CS445\cs445-lab1

Mode                LastWriteTime         Length Name
----                -
-a----             4/27/2022   9:25 AM             23 index1.js
-a----             4/27/2022   8:55 AM             34 README.md

PS D:\MIU\CS445\CS445\cs445-lab1> git add .
PS D:\MIU\CS445\CS445\cs445-lab1> git commit -m "index1.js"
[main 7a19096] index1.js
 1 file changed, 1 insertion(+)
 create mode 100644 index1.js
PS D:\MIU\CS445\CS445\cs445-lab1> 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), 298 bytes | 149.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ngoctamnguyen/CS445-lab1
 d1286f7..7a19096 main -> main
PS D:\MIU\CS445\CS445\cs445-lab1>
```

5. git fetch

The screenshot shows the VS Code interface with a file explorer on the left and a terminal on the right. The file explorer shows a project named 'CS445' with a subdirectory 'CS445-lab1'. The 'README.md' file is selected. The terminal shows the following commands and output:

```
CS445-lab1 > README.md > # CS445-lab1
1 # CS445-lab1
2 This is CS445 lab1
3
4 This line was edited on github
5
6 This line is use to test git fetch:
7 (on local use: git fetch, then git merge)
8

PS D:\MIU\CS445\CS445\cs445-lab1> git fetch
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), 761 bytes | 47.00 KiB/s, done.
From https://github.com/ngoctamnguyen/CS445-lab1
  7bab7b1..3d6094f  main    -> origin/main
PS D:\MIU\CS445\CS445\cs445-lab1> git merge
Updating 7bab7b1..3d6094f
Fast-forward
 README.md | 3 +++
 1 file changed, 3 insertions(+)
PS D:\MIU\CS445\CS445\cs445-lab1>
```

6. git pull

The screenshot shows the VS Code interface with a file explorer on the left and a terminal on the right. The file explorer shows the same project as the previous screenshot. The terminal shows the following commands and output:

```
PS D:\MIU\CS445\CS445\cs445-lab1> git pull
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (6/6), 1.37 KiB | 46.00 KiB/s, done.
From https://github.com/ngoctamnguyen/CS445-lab1
  7a19096..7bab7b1  main    -> origin/main
Updating 7a19096..7bab7b1
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
PS D:\MIU\CS445\CS445\cs445-lab1>
```

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



4. Clone your forked repository into your local machine

```
PS D:\MIU\CS445\CS445\cs445-lab1> git clone https://github.com/ngoctamnguyen/CS445-lab1.git
Cloning into 'CS445-lab1'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 18 (delta 1), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (18/18), done.
Resolving deltas: 100% (1/1), done.
PS D:\MIU\CS445\CS445\cs445-lab1> |
```

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

```
PS D:\MIU\CS445\CS445\cs445-lab1> git branch 610001-JohnSmith
PS D:\MIU\CS445\CS445\cs445-lab1> git branch
 610001-JohnSmith
* main
PS D:\MIU\CS445\CS445\cs445-lab1> |
```

6. Checkout your new branch

```
PS D:\MIU\CS445\CS445\cs445-lab1> git checkout 610001-JohnSmith
Switched to branch '610001-JohnSmith'
M      CS445-lab1
PS D:\MIU\CS445\CS445\cs445-lab1> |
```

7. Add a new file with your name, example: hello.js
8. Commit the changes

```
PS D:\MIU\CS445\CS445\cs445-lab1> git commit -m "commit in 610001-JohnSmith"
[610001-JohnSmith c6b434b] commit in 610001-JohnSmith
1 file changed, 1 insertion(+)
create mode 160000 CS445-lab1
```

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

```

PS D:\MIU\CS445\CS445\cs445-lab1> git branch
610001-JohnSmith
* main
PS D:\MIU\CS445\CS445\cs445-lab1> git merge 610001-JohnSmith
Updating 3d6094f..c6b434b
Fast-forward
 CS445-lab1 | 1 +
 1 file changed, 1 insertion(+)
 create mode 160000 CS445-lab1
PS D:\MIU\CS445\CS445\cs445-lab1>

```

10. Push your changes to the remote repo

```

PS D:\MIU\CS445\CS445\cs445-lab1> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 537 bytes | 268.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/ngoctamnguyen/CS445-lab1
 f6288bb..60ceb3e main -> main
PS D:\MIU\CS445\CS445\cs445-lab1>

```

 ngoctamnguyen Merge branch '610001-JohnSmith'			60ceb3e 2 minutes ago	🕒 10 commits
	CS445-lab1	commit in 610001-JohnSmith	14 minutes ago	
	README.md	Update README.md	6 hours ago	
	index1.js	Update index1.js	26 minutes ago	
	lab1.js	committed in 610001-JohnSmith	2 minutes ago	