

# Embedded computing for scientific and industrial imaging applications

---

Lecture 3 - Git and GitHub demo

# Outline

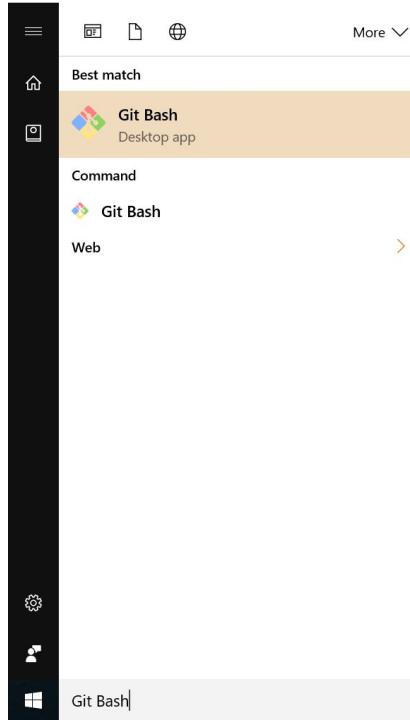
- Git demo
  - My first Git and GitHub repository
  - Branching, merging and resolving conflicts.
- GitHub demo
  - Fork
  - Pull request
- C demo
- Computing square roots

# Git demo - My first git repository

---

# Git Demo - Git Bash

- Press Windows key and type “Git Bash” and Execute



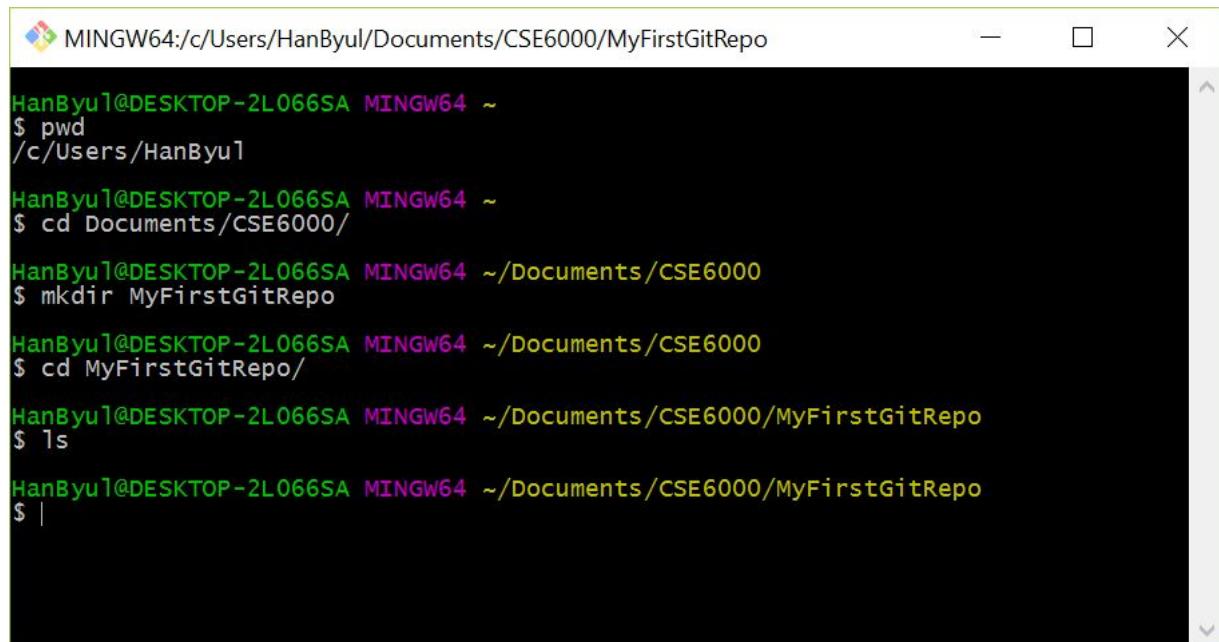
The image shows a screenshot of a Git Bash terminal window. The title bar indicates the session is running on "MINGW64:/c/Users/HanByul". The command prompt shows "HanByul@DESKTOP-2L066SA MINGW64 ~" followed by a dollar sign (\$) and a vertical bar (|). The main area of the terminal is completely black, indicating it is currently empty or has no output to display.

# Git demo - shell command

- \$ pwd - print working directory
- \$ cd - change directory
- \$ mkdir - make directory
- \$ ls - list
- \$ mv - move
- \$ cp - copy
- \$ clear - clear

See more

[UW AMATH 483/583 : Shell](#)



The screenshot shows a terminal window titled 'MINGW64:/c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo'. The window contains the following command history:

```
HanByu1@DESKTOP-2L066SA MINGW64 ~
$ pwd
/c/Users/HanByu1

HanByu1@DESKTOP-2L066SA MINGW64 ~
$ cd Documents/CSE6000

HanByu1@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000
$ mkdir MyFirstGitRepo

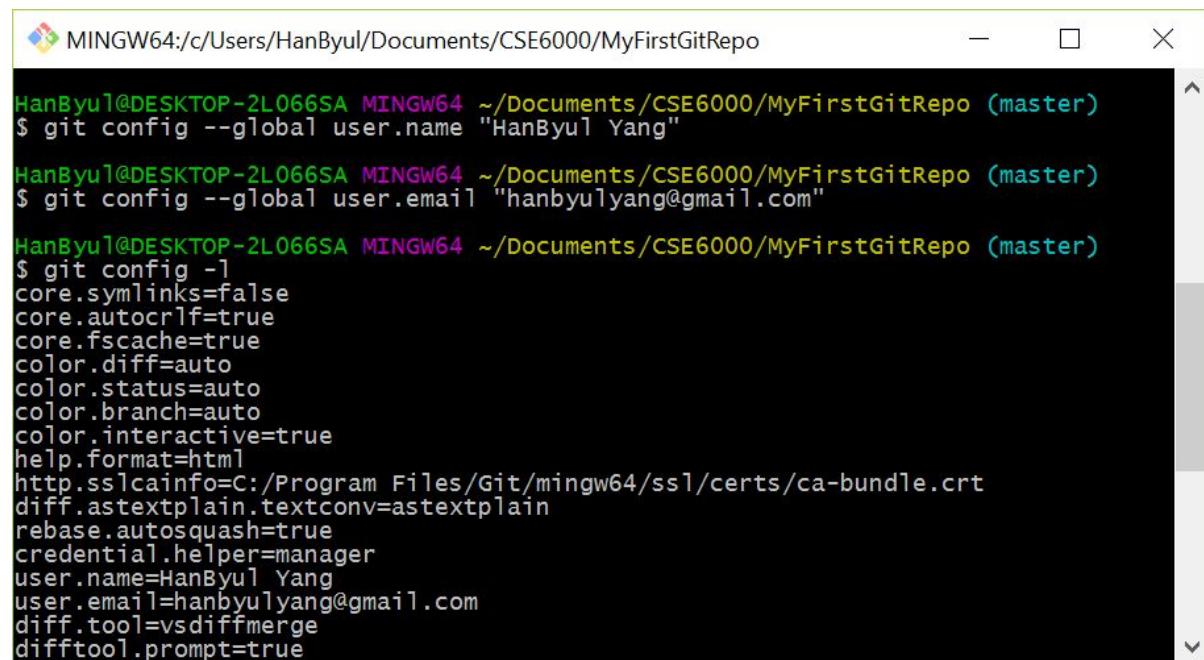
HanByu1@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000
$ cd MyFirstGitRepo/

HanByu1@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo
$ ls

HanByu1@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo
$ |
```

# Git demo - git config

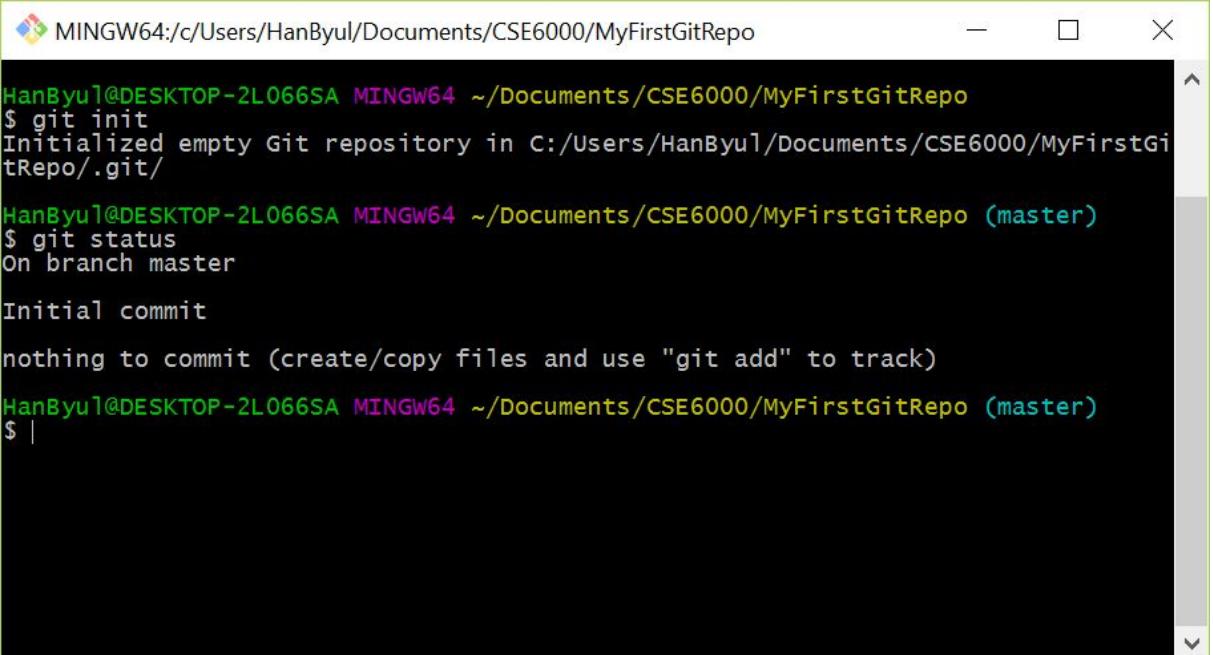
- \$ git config --global user.name "name"
- \$ git config --global user.email "email"
- \$ git config -l



```
MINGW64:/c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git config --global user.name "HanByul Yang"
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git config --global user.email "hanbyulyang@gmail.com"
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git config -l
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
credential.helper=manager
user.name=HanByul Yang
user.email=hanbyulyang@gmail.com
diff.tool=vsdiffmerge
difftool.prompt=true
```

# Git demo - git init

- \$ git init - initialize empty git repository
- \$ git status - check current status of git repository



```
MINGW64:/c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo
$ git init
Initialized empty Git repository in C:/Users/HanByul/Documents/CSE6000/MyFirstGitRepo/.git/
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git status
On branch master

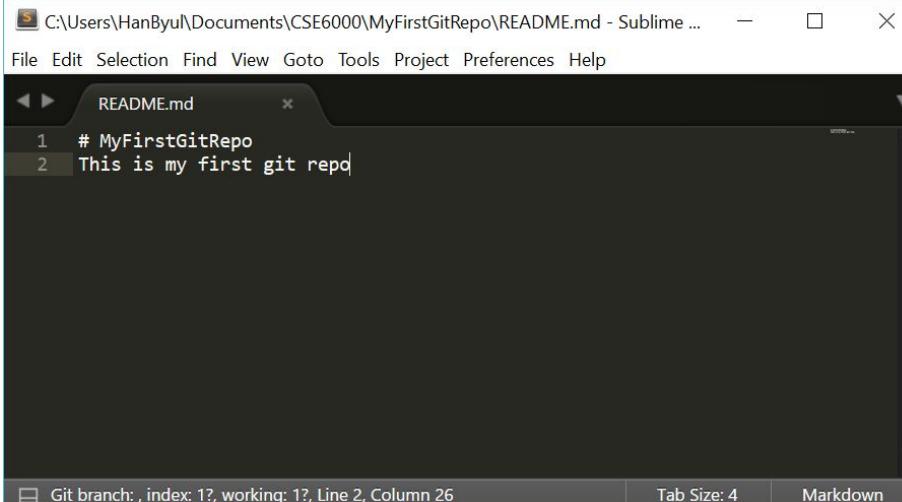
Initial commit

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

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ |
```

# Git demo - git add

- Create a file “README.md” with any text editor (e.g. notepad, sublime text)
- \$ git add - add files to staging area to commit
- \$ git status



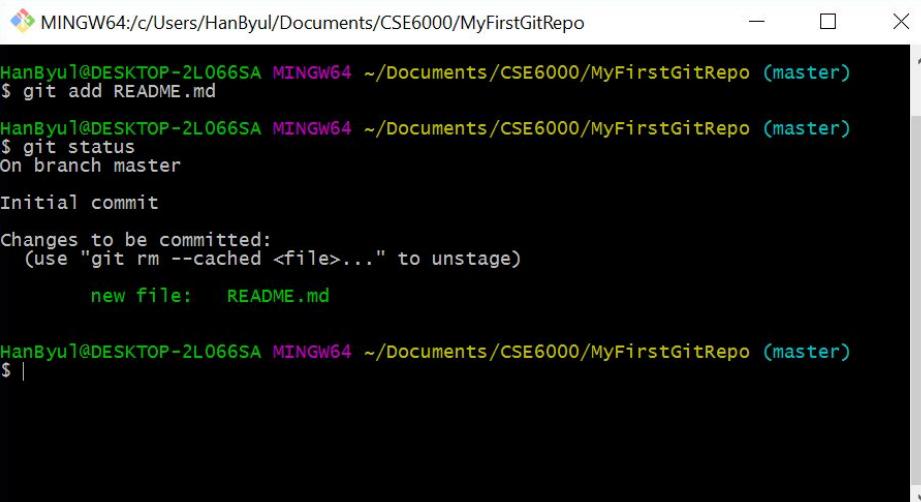
C:\Users\HanByul\Documents\CSE6000\MyFirstGitRepo\README.md - Sublime ...

File Edit Selection Find View Goto Tools Project Preferences Help

README.md

```
1 # MyFirstGitRepo
2 This is my first git repd
```

Git branch: , index: 1?, working: 1?, Line 2, Column 26 | Tab Size: 4 | Markdown



MINGW64:/c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git add README.md

HabByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git status
On branch master

Initial commit

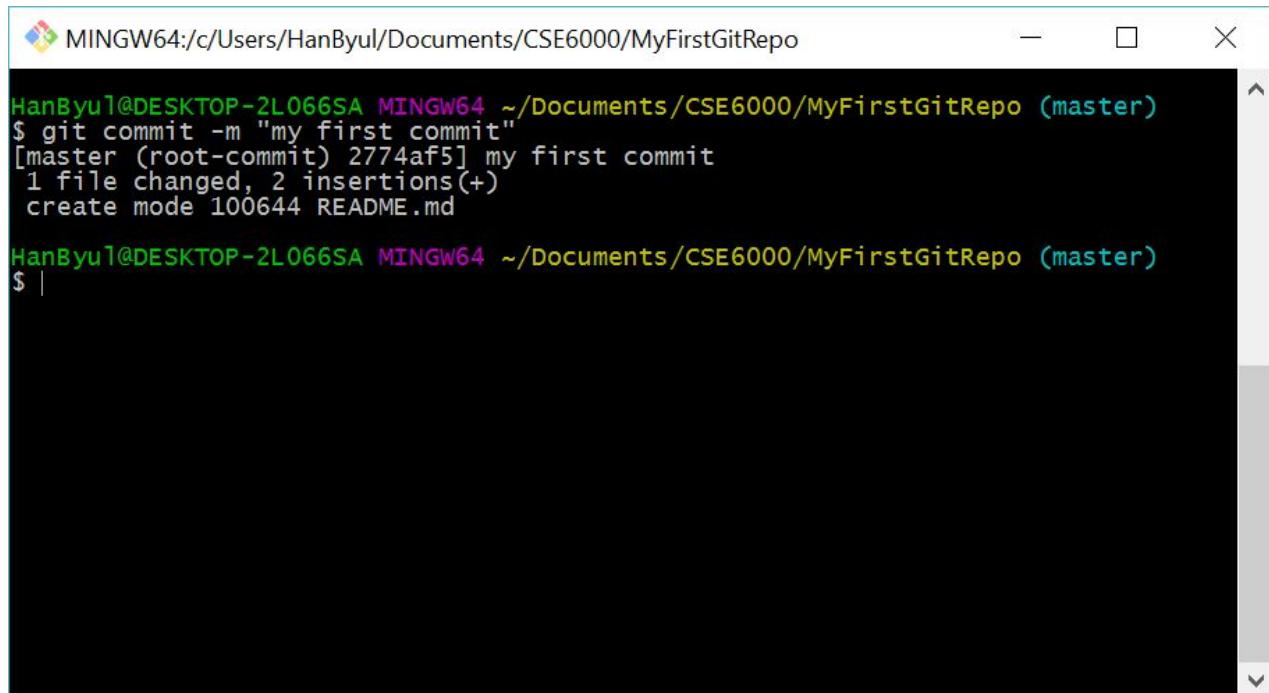
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   README.md

HabByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ |
```

# Git demo - git commit

- \$ git commit -m "my first commit" - commit with some comments



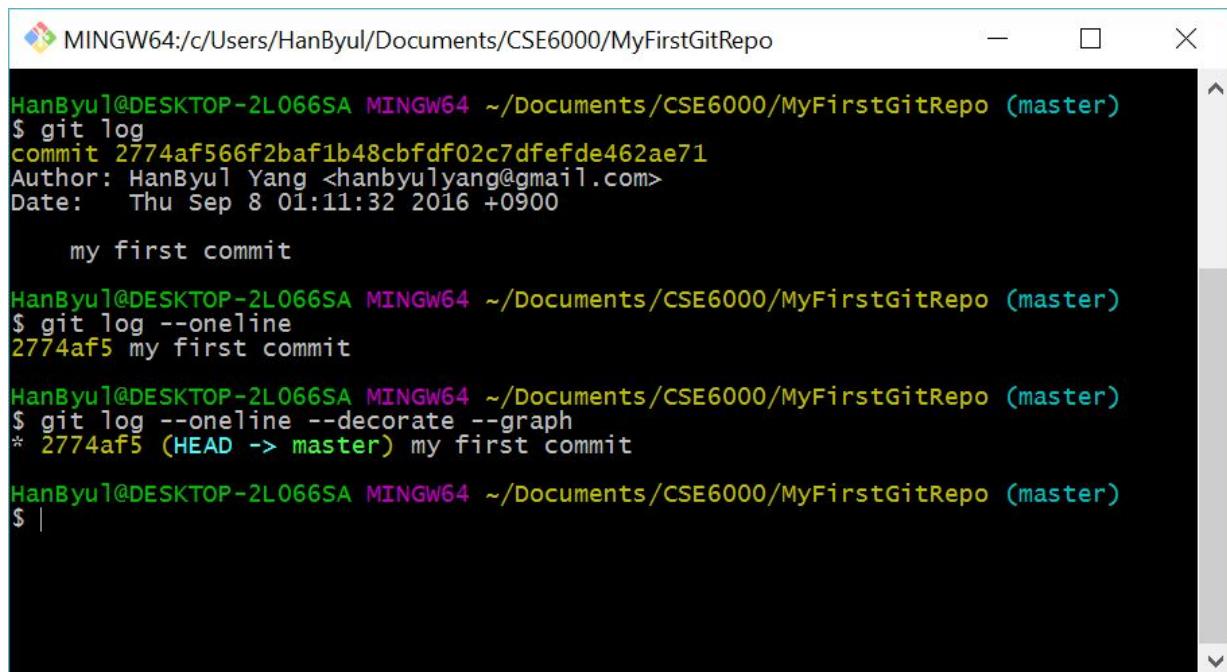
A screenshot of a terminal window titled "MINGW64:c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo". The window shows the command \$ git commit -m "my first commit" being run, followed by the output of the commit: [master (root-commit) 2774af5] my first commit, 1 file changed, 2 insertions(+), and a file named README.md created with mode 100644. The terminal prompt \$ | is visible at the bottom.

```
MINGW64:c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo (master)
$ git commit -m "my first commit"
[master (root-commit) 2774af5] my first commit
 1 file changed, 2 insertions(+)
 create mode 100644 README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ |
```

# Git demo - git log

- \$ git log - show history of commits



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo (master)". The window displays the following command and its output:

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git log
commit 2774af566f2baf1b48cbfdf02c7dfefde462ae71
Author: HanByul Yang <hanbyulyang@gmail.com>
Date:   Thu Sep 8 01:11:32 2016 +0900

    my first commit

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git log --oneline
2774af5 my first commit

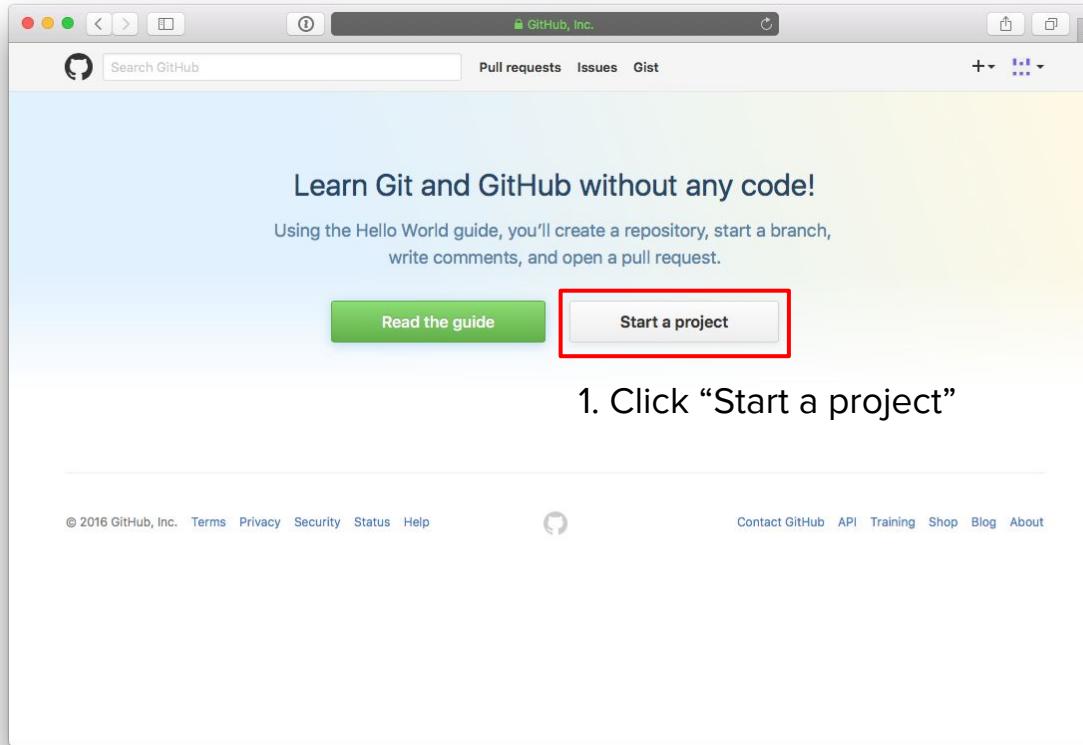
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git log --oneline --decorate --graph
* 2774af5 (HEAD -> master) my first commit

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ |
```

# GitHub demo - My first git repository

---

# GitHub demo - Start a project



# GitHub demo - Create a new repository

The screenshot shows the GitHub 'Create a new repository' interface. At the top, there's a navigation bar with links for 'Pull requests', 'Issues', and 'Gist'. Below that, the main title is 'Create a new repository' with a subtitle: 'A repository contains all the files for your project, including the revision history.' On the left, there's a 'Owner' dropdown set to 'CSE6000'. The 'Repository name' field contains 'MyFirstGitRepo' and has a green checkmark icon to its right. A red box highlights this field. Below it, there's a description input field and a radio button for 'Public' repositories, which is selected. Another red box highlights this radio button. There's also an option for 'Private' repositories. At the bottom, there's a checkbox for 'Initialize this repository with a README' and two buttons: 'Add .gitignore: None' and 'Add a license: None'. A red box highlights the large green 'Create repository' button at the bottom.

1. Type repository name “MyFirstGitRepo”
2. Click “Create repository”

# GitHub demo - push an existing repository

The screenshot shows a GitHub repository page for 'CSE6000 / MyFirstGitRepo'. The page includes sections for adding collaborators, quick setup instructions, command-line creation steps, and a highlighted section for pushing an existing repository.

**Give access to the people you work with**  
You should give access to the collaborators and teams you need to work with.  
[Add teams and collaborators](#)

**Quick setup — if you've done this kind of thing before**  
Set up in Desktop or HTTPS SSH <https://github.com/CSE6000/MyFirstGitRepo.git>  
We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

**...or create a new repository on the command line**

```
echo "# MyFirstGitRepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/CSE6000/MyFirstGitRepo.git
git push -u origin master
```

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/CSE6000/MyFirstGitRepo.git
git push -u origin master
```

Type commands on Git Bash

# GitHub demo - git push



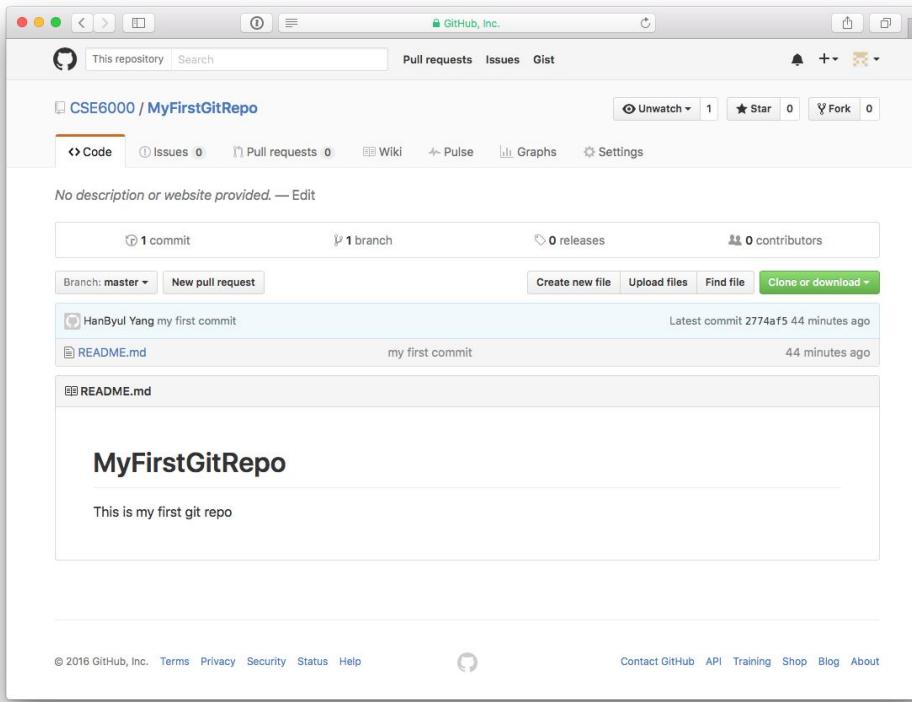
MINGW64:/c/Users/HanByul/Documents/CSE6000/MyFirstGitRepo

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git remote add origin https://github.com/CSE6000/MyFirstGitRepo.git

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ git push -u origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 256 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/CSE6000/MyFirstGitRepo.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/CSE6000/MyFirstGitRepo (master)
$ |
```

# GitHub demo

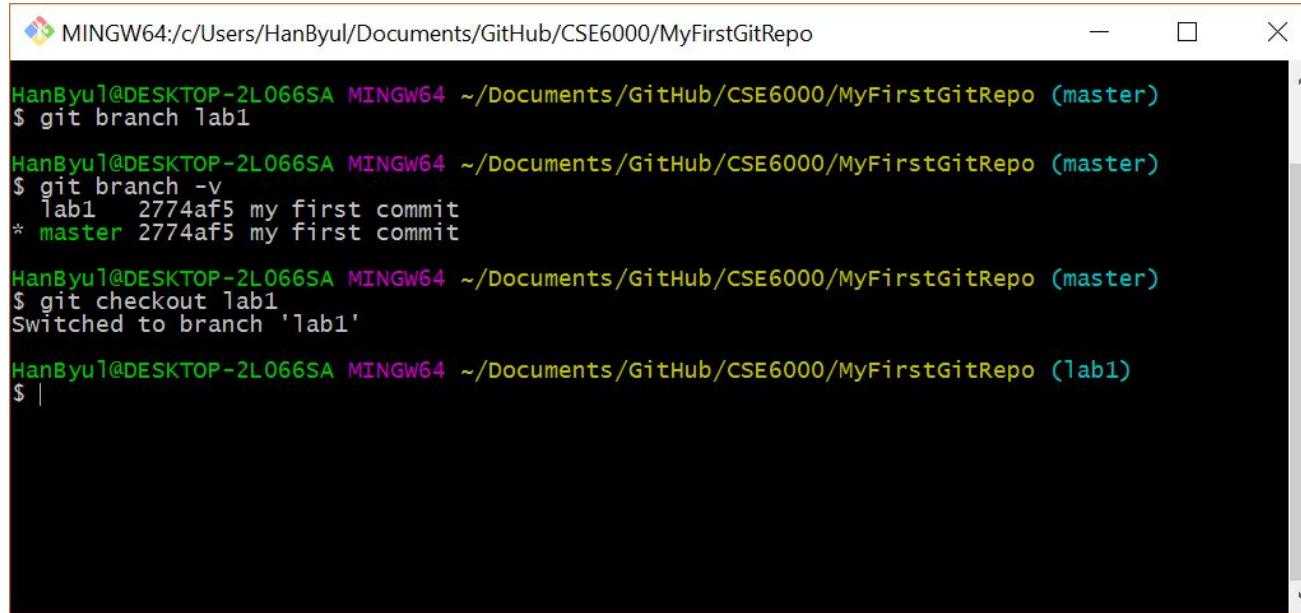


# Git demo - Branch and merge

---

# Git demo - branch

- \$ git branch lab1 - create branch “lab1”
- \$ git branch -v - show branches
- \$ git checkout lab1 - Switch branch to lab1



```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git branch lab1

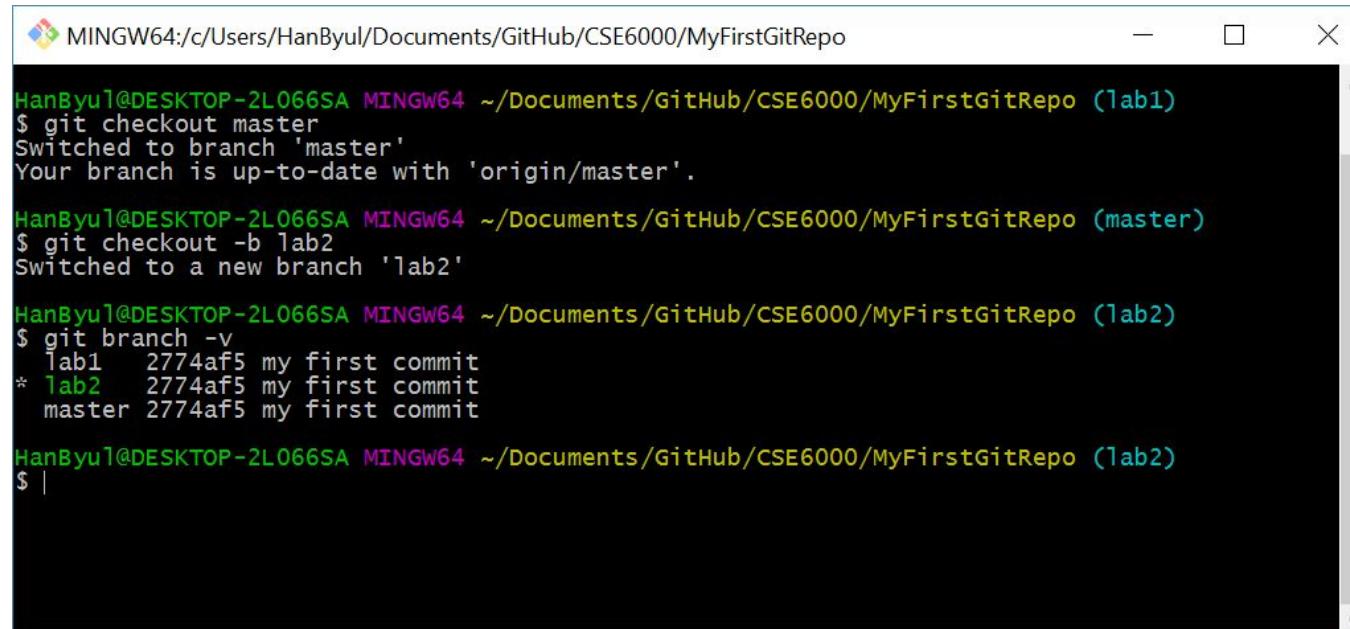
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git branch -v
  lab1 2774af5 my first commit
* master 2774af5 my first commit

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git checkout lab1
Switched to branch 'lab1'

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ |
```

# Git demo - branch

- \$ git checkout master - switch branch to master
- \$ git checkout -b lab2 - create and switch branch lab2
- \$ git branch -v - show branches



```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

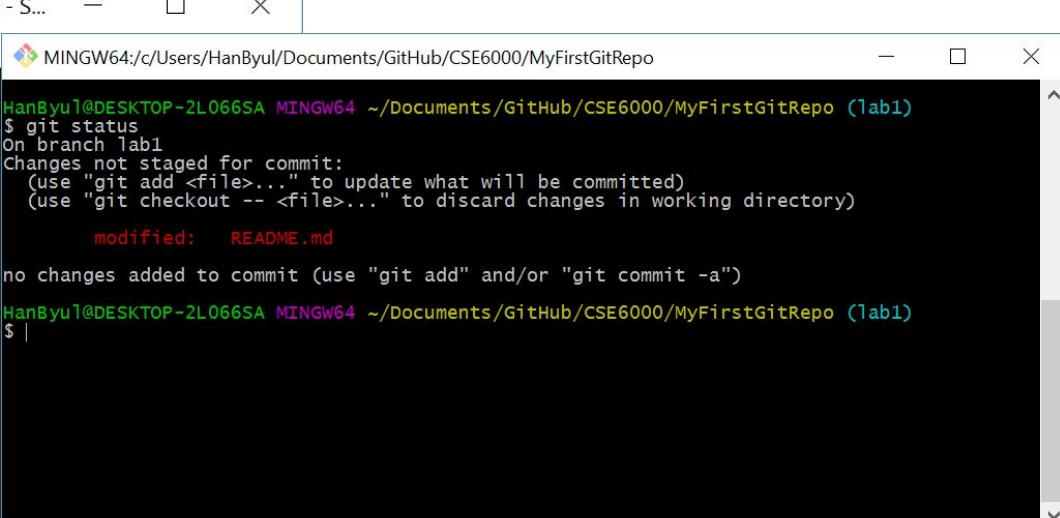
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git checkout -b lab2
Switched to a new branch 'lab2'

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab2)
$ git branch -v
  lab1  2774af5 my first commit
* 1ab2  2774af5 my first commit
  master 2774af5 my first commit

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab2)
$ |
```

# Git demo - branch

- \$ git checkout lab1 - switch to lab1
- Edit “README.md” as below. (added Lab1 section)
- \$ git status



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo". The window displays the following text:

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git status
On branch lab1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ |
```

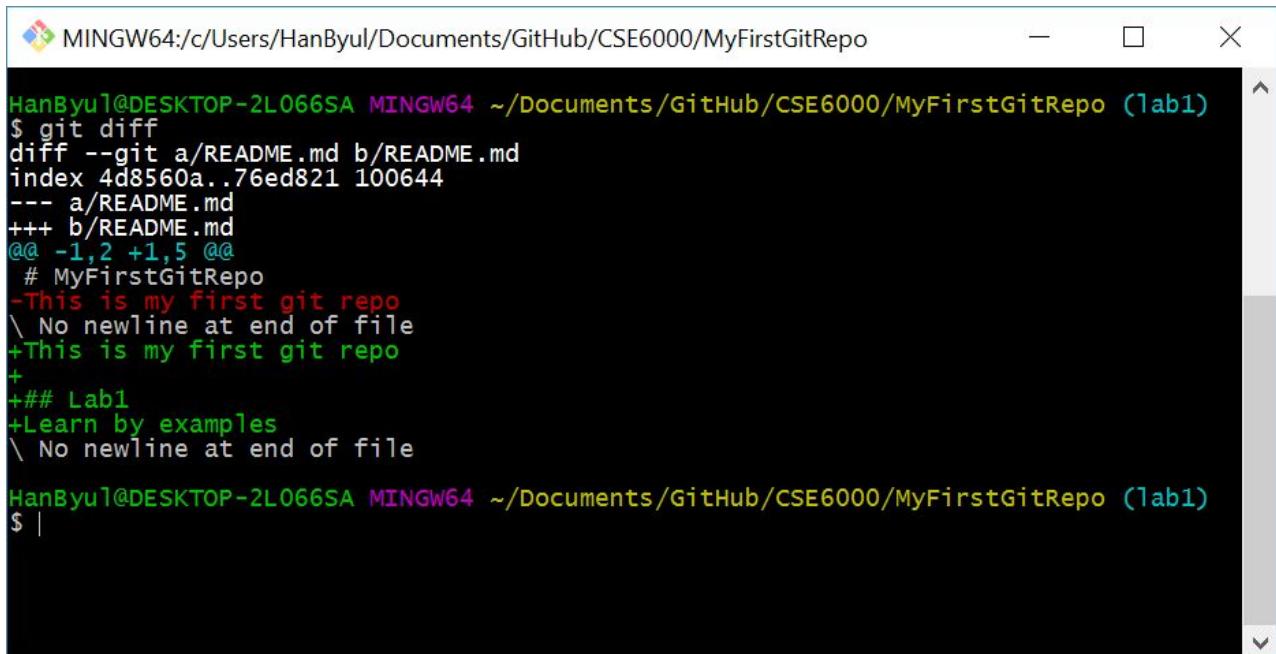
Below the terminal window, there is a code editor window titled "README.md". The code editor shows the following content:

```
1 # MyFirstGitRepo
2 This is my first git repo
3
4 ## Lab1
5 Learn by examples|
```

The status bar at the bottom of the code editor indicates: "Git branch: lab1, index: 0, working: 1#, Line 5, Column 18".

# Git demo - diff

- \$ git diff - show differences between working directory and most recent commit



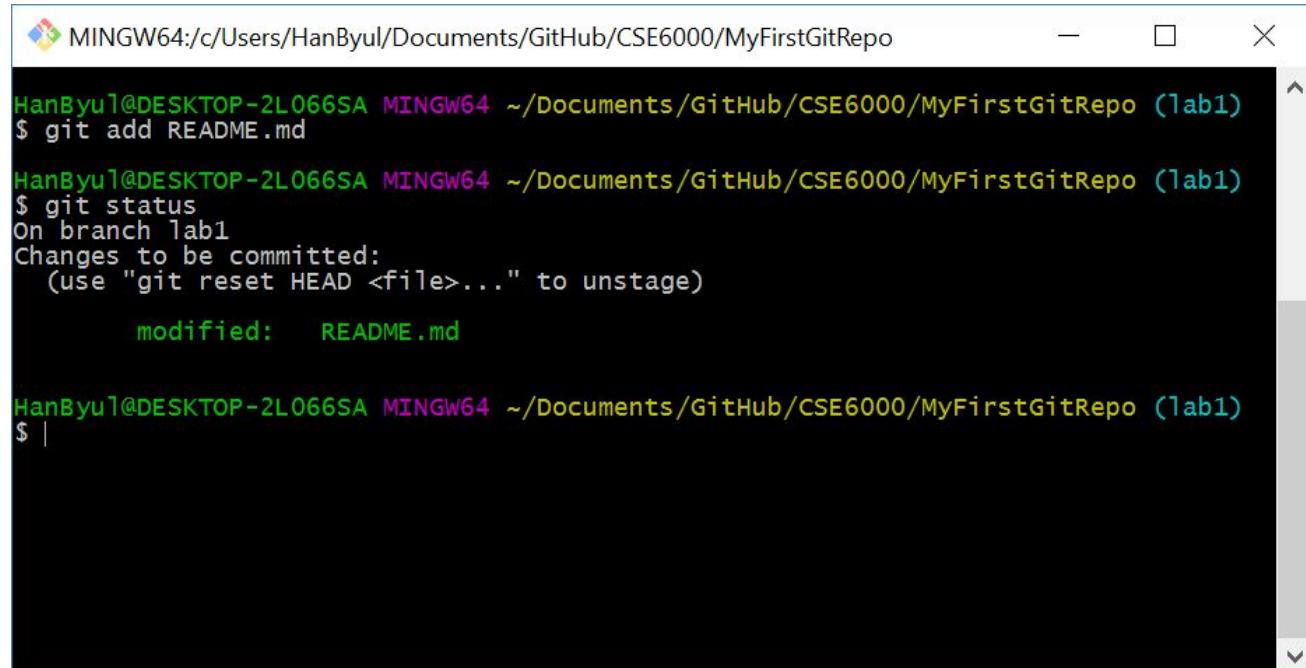
The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)". The command \$ git diff is run, displaying a diff output comparing two versions of the README.md file. The changes are as follows:

```
diff --git a/README.md b/README.md
index 4d8560a..76ed821 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,5 @@
 # MyFirstGitRepo
-This is my first git repo
\ No newline at end of file
+This is my first git repo
+
+## Lab1
+Learn by examples
\ No newline at end of file
```

The terminal prompt HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1) is shown again, followed by a dollar sign.

# Git demo - diff

- \$ git add README.md
- \$ git status



```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git add README.md

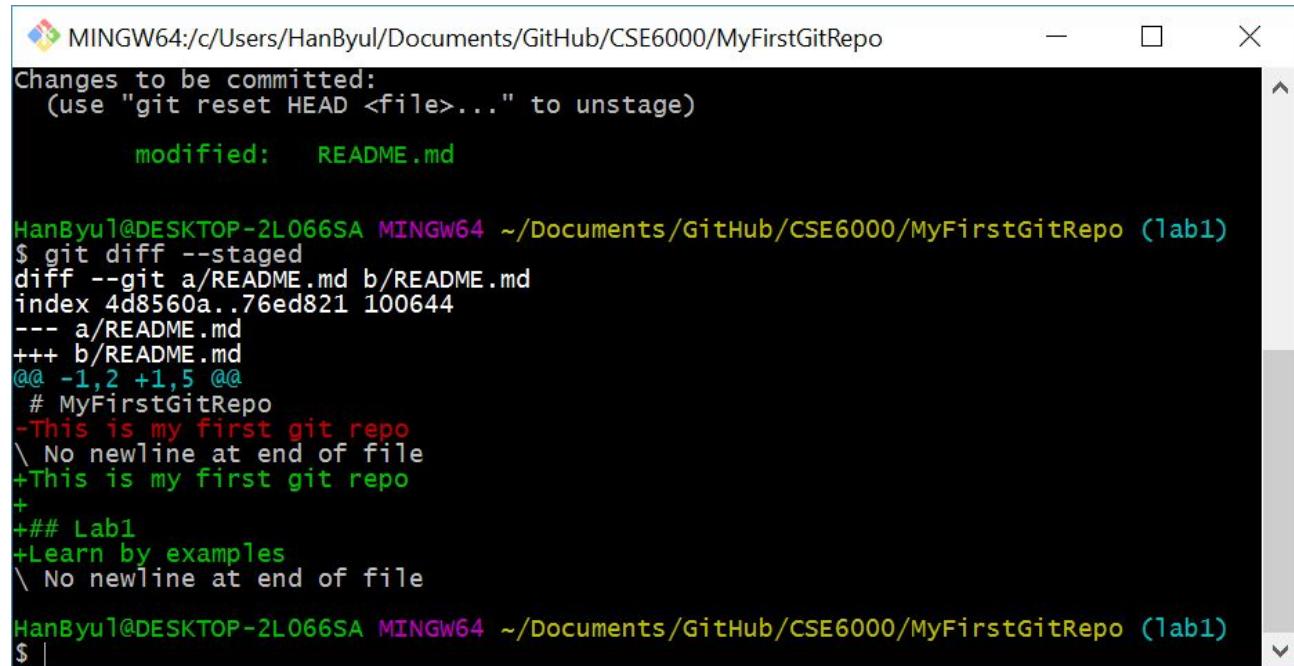
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git status
On branch lab1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ |
```

# Git demo - diff

- \$ git diff --staged



```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

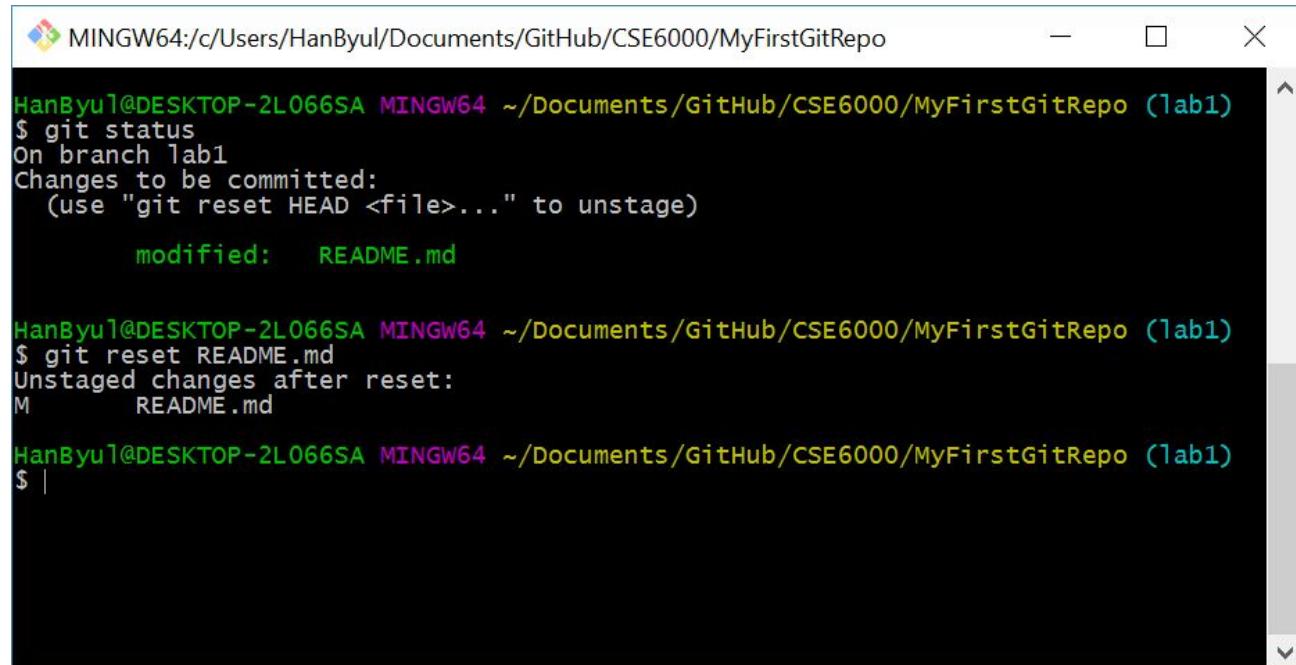
        modified:   README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git diff --staged
diff --git a/README.md b/README.md
index 4d8560a..76ed821 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,5 @@
 # MyFirstGitRepo
-This is my first git repo
\ No newline at end of file
+This is my first git repo
+
+## Lab1
+Learn by examples
\ No newline at end of file

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ |
```

# Git demo - git reset

- \$ git reset README.md - Unstage README.md



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo". The terminal output is as follows:

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git status
On branch lab1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

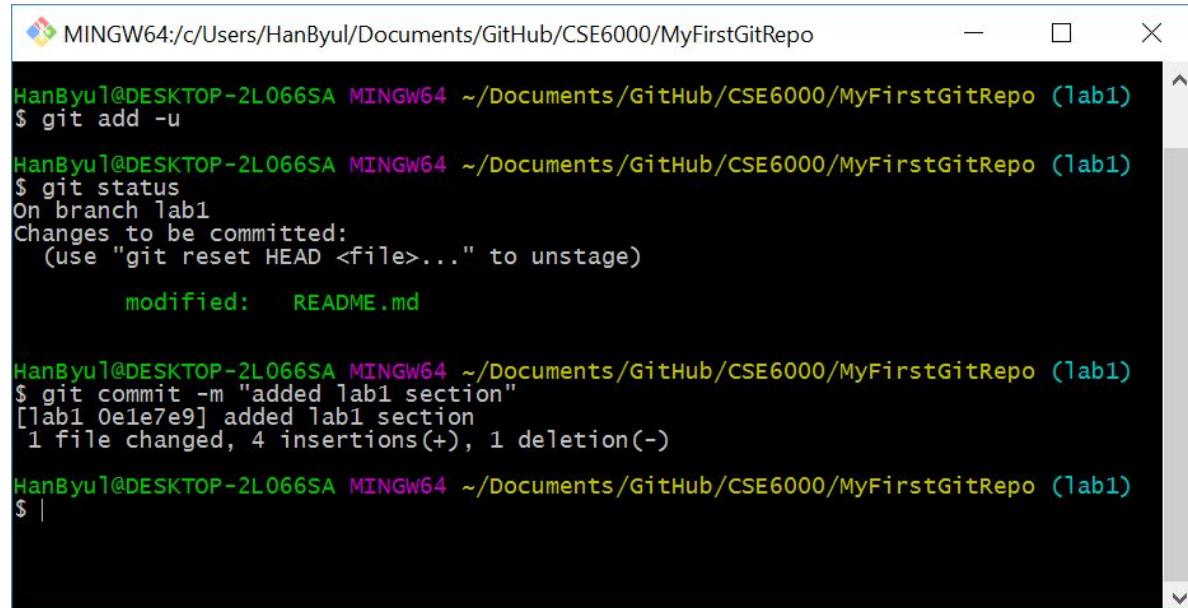
    modified:   README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git reset README.md
Unstaged changes after reset:
M       README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ |
```

The terminal shows the user running `git status`, which indicates a modified file `README.md`. Then, the user runs `git reset README.md`, which removes the file from the staging area, as shown by the `M` status in the second command's output.

- \$ git add -u - stage all changes of tracked files
- \$ git commit -m "added lab1 section"



```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git add -u

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git status
On branch lab1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

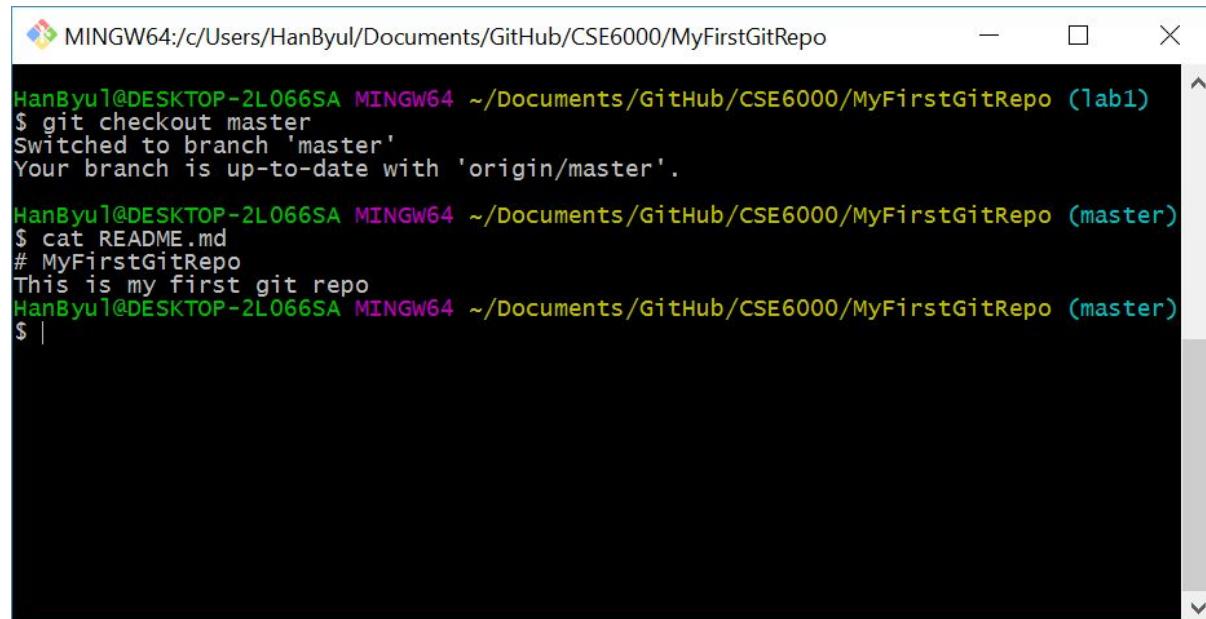
        modified:   README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git commit -m "added lab1 section"
[lab1 0e1e7e9] added lab1 section
 1 file changed, 4 insertions(+), 1 deletion(-)

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ |
```

# Git demo

- \$ git checkout master
- \$ cat README.md - Concatenate and print (display) the contents of files.



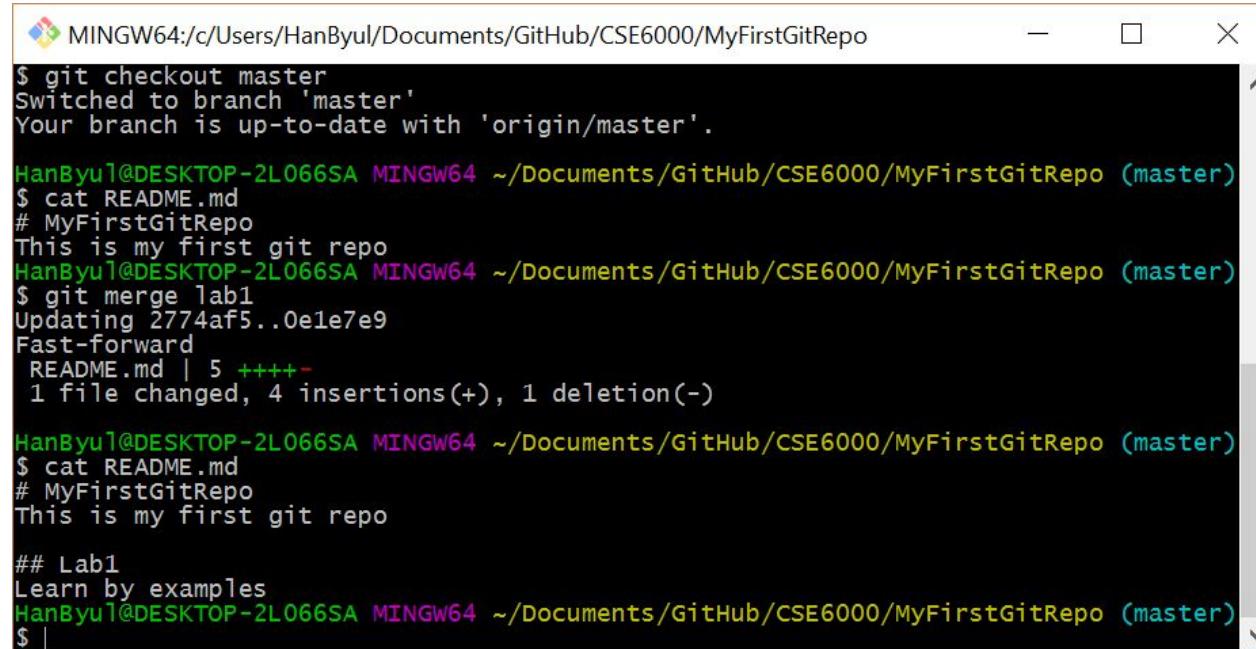
A screenshot of a terminal window titled "MINGW64:c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo". The window shows the following command-line session:

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab1)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ cat README.md
# MyFirstGitRepo
This is my first git repo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ |
```

# Git demo - merge

- \$ git merge lab1 - merge commits of lab1 to current branch (master)



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo". The terminal output is as follows:

```
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.

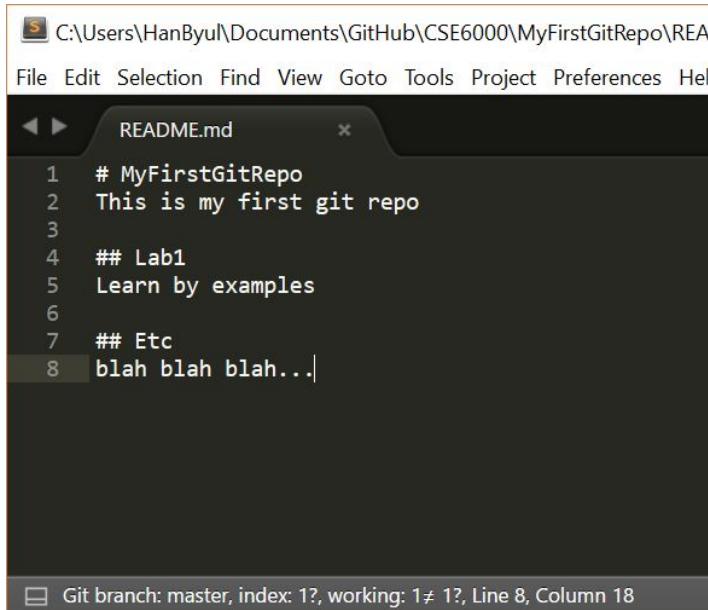
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ cat README.md
# MyFirstGitRepo
This is my first git repo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git merge lab1
Updating 2774af5..0e1e7e9
Fast-forward
 README.md | 5 +---+
 1 file changed, 4 insertions(+), 1 deletion(-)

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ cat README.md
# MyFirstGitRepo
This is my first git repo

## Lab1
Learn by examples
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ |
```

# Git demo - merge and conflict

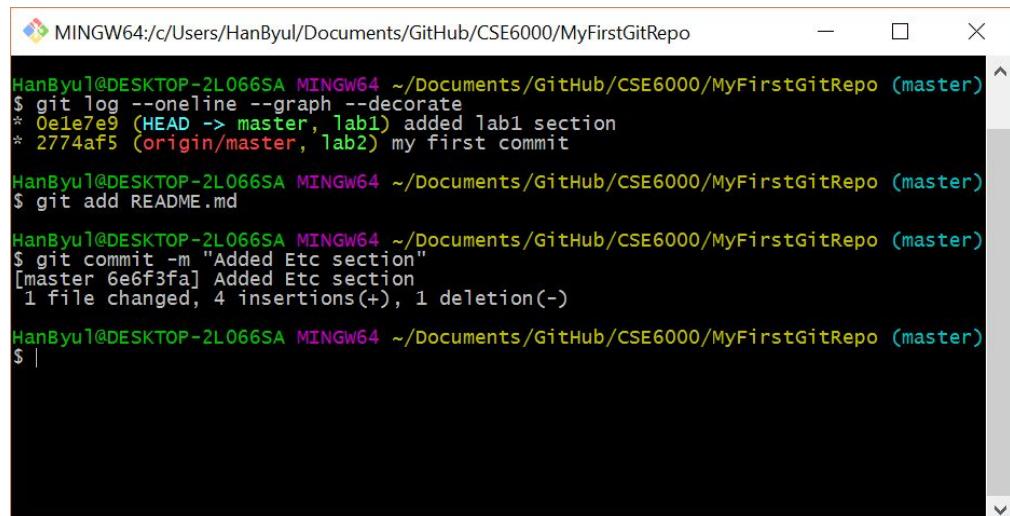
- Edit “README.md” and commit on master branch



The screenshot shows a dark-themed code editor window. The title bar reads "C:\Users\HanByul\Documents\GitHub\CSE6000\MyFirstGitRepo\README.md". The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. Below the menu is a toolbar with icons for back, forward, and close. The main area displays the following content:

```
1 # MyFirstGitRepo
2 This is my first git repo
3
4 ## Lab1
5 Learn by examples
6
7 ## Etc
8 blah blah blah...
```

A status bar at the bottom indicates "Git branch: master, index: 1?, working: 1≠ 1?, Line 8, Column 18".



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo". The command \$ git log --oneline --graph --decorate is run, showing two commits:

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git log --oneline --graph --decorate
* 0e1e7e9 (HEAD -> master, lab1) added lab1 section
* 2774af5 (origin/master, lab2) my first commit
```

The command \$ git add README.md is run.

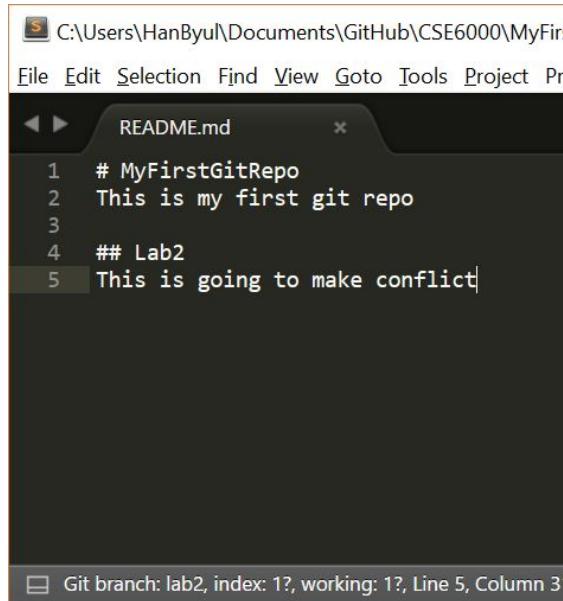
```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git add README.md
```

The command \$ git commit -m "Added Etc section" is run, resulting in a new commit [master 6e6f3fa] Added Etc section.

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git commit -m "Added Etc section"
[master 6e6f3fa] Added Etc section
 1 file changed, 4 insertions(+), 1 deletion(-)
```

The final command \$ | is shown at the bottom.

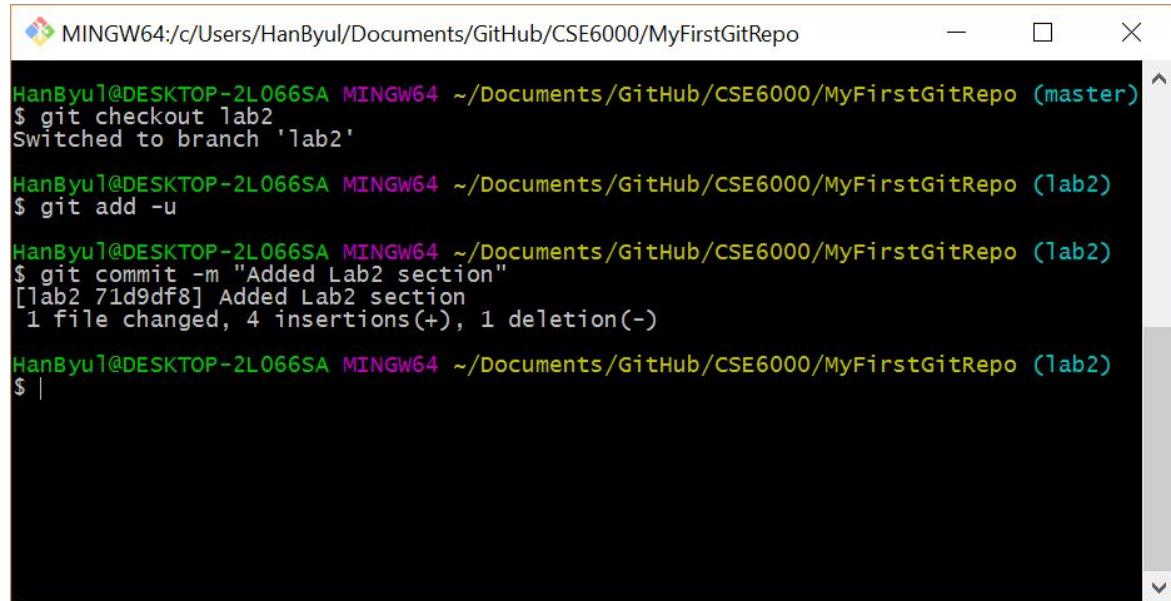
# Git demo - merge and conflict



C:\Users\HanByul\Documents\GitHub\CSE6000\MyFirstGitRepo

```
File Edit Selection Find View Goto Tools Project Project Properties
  README.md
  1 # MyFirstGitRepo
  2 This is my first git repo
  3
  4 ## Lab2
  5 This is going to make conflict
```

Git branch: lab2, index: 1?, working: 1?, Line 5, Column 31



MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git checkout lab2
Switched to branch 'lab2'

HannByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab2)
$ git add -u

HannByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab2)
$ git commit -m "Added Lab2 section"
[lab2 71d9df8] Added Lab2 section
1 file changed, 4 insertions(+), 1 deletion(-)

HannByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab2)
$ |
```

# Git demo - merge and conflict

```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (lab2)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git merge lab2
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master | MERGING)
$ |
```

```
C:\Users\HanByul\Documents\GitHub\CSE6000\MyFirstGitRepo
File Edit Selection Find View Goto Tools Project Preferences
README.md
1 # MyFirstGitRepo
2 This is my first git repo
3
4 <<<<< HEAD
5 ## Lab1
6 Learn by examples| ←
7
8 ## Etc
9 blah blah blah...
10 =====
11 ## Lab2
12 This is going to make conflict
13 >>>> lab2
14

Git branch: master, index: 1? 1!!, working: 1? 1!!, Line 6, C
```

# Git demo - merge and conflict

```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master|MERGING)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified: README.md

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

no changes added to commit (use "git add" and/or "git commit -a")

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master|MERGING)
$ |
```

# Git demo - merge and conflict

C:\Users\HanByul\Documents\GitHub\CSE6000\MyFirstGitRepo\README.md

File Edit Selection Find View Goto Tools Project Preferences Help

```
1 # MyFirstGitRepo
2 This is my first git repo
3
4 <<<<< HEAD
5 ## Lab1
6 Learn by examples
7
8 ## Etc
9 blah blah blah...
10 =====
11 ## Lab2
12 This is going to make conflict
13 >>>>> lab2
14
```

Git branch: master, index: 1? 1!!, working: 1? 1!!, Line 6, C

C:\Users\HanByul\Documents\GitHub\CSE6000\MyFirstGitRepo\README.md - S...

File Edit Selection Find View Goto Tools Project Preferences Help

```
1 # MyFirstGitRepo
2 This is my first git repo
3
4 ## Lab1
5 Learn by examples
6
7 ## Lab2
8 This is going to make conflict
9
10 ## Etc
11 blah blah blah...
12
13
```

Git branch: master, index: 1? 1??, working: 1?, Line 8, Column 31

Tab Size: 4

Mark

# Git demo - merge and conflict

```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo ━ ━ X ↻
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master|MERGING)
$ git add README.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master|MERGING)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:

  modified:   README.md

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

    test.txt

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master|MERGING)
$ |
```

# Git demo - merge and conflict

```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master|MERGING)
$ git commit
[master df92f69] Merge branch 'lab2'
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ |
```

```
MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo
Merge branch 'lab2'

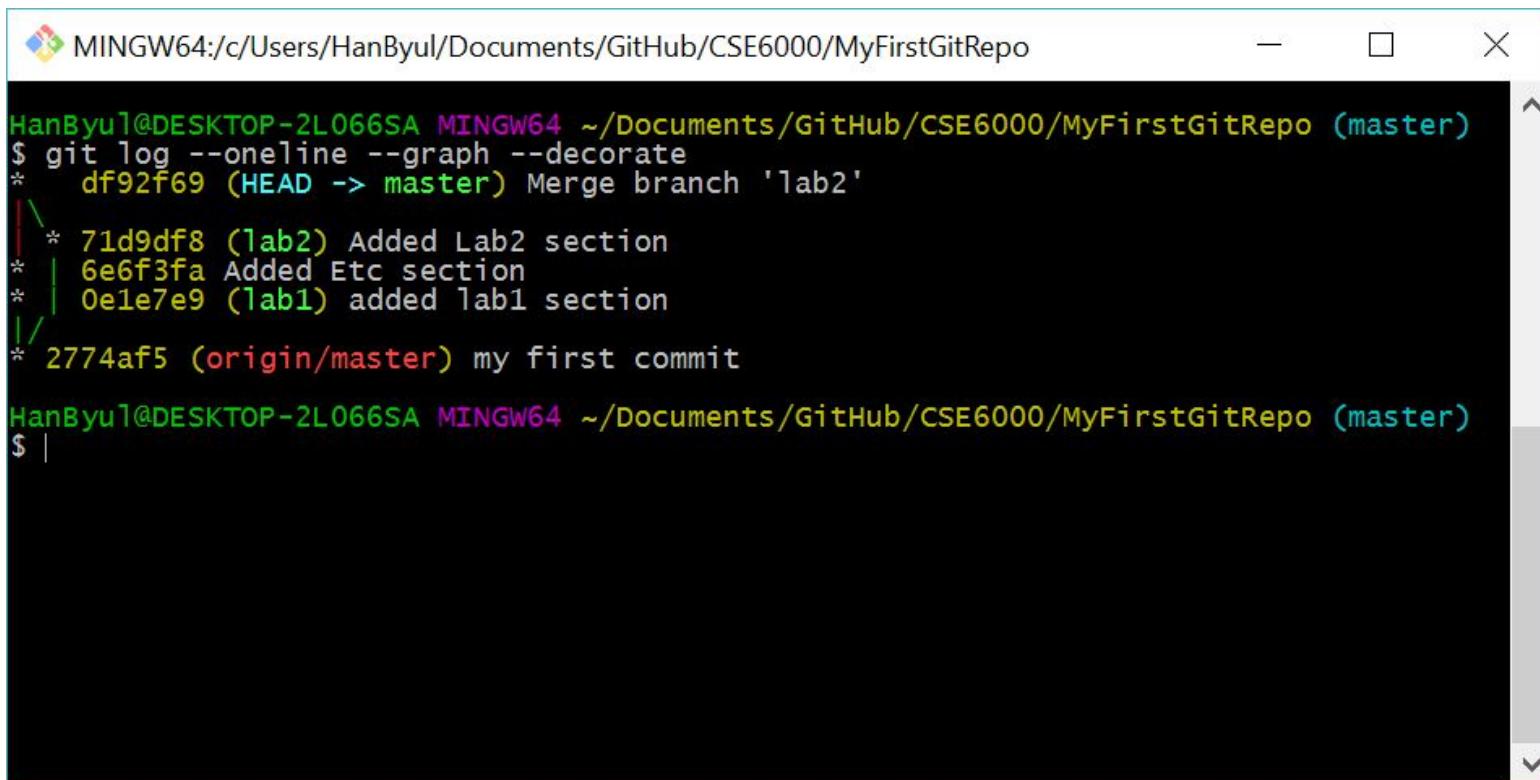
# Conflicts:
#       README.md
#
# It looks like you may be committing a merge.
# If this is not correct, please remove the file
#       .git/MERGE_HEAD
# and try again.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#   (use "git push" to publish your local commits)
#
# All conflicts fixed but you are still merging.
#
# Changes to be committed:
#       modified:   README.md
#
<documents/GitHub/CSE6000/MyFirstGitRepo/.git/COMMIT_EDITMSG [unix] (23:57 08/09/2016)1,1 ALL
<sers/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo/.git/COMMIT_EDITMSG" [unix] 22L, 548C
```

It is vi editor.

Type keys below  
ESC, :, w, q  
(save and exit)

# Git demo



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/GitHub/CSE6000/MyFirstGitRepo". The window displays a Git log output from the master branch. The log shows the following commits:

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)
$ git log --oneline --graph --decorate
*   df92f69 (HEAD -> master) Merge branch 'lab2'
|\ 
*   71d9df8 (lab2) Added Lab2 section
*   | 6e6f3fa Added Etc section
*   | 0e1e7e9 (lab1) added lab1 section
|/
* 2774af5 (origin/master) my first commit
```

Below the log, there is a prompt: "HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/GitHub/CSE6000/MyFirstGitRepo (master)" followed by a dollar sign (\$) and a blank line.

# Git

<http://xkcd.com/1597/>

THIS IS GIT. IT TRACKS COLLABORATIVE WORK  
ON PROJECTS THROUGH A BEAUTIFUL  
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZIZE THESE SHELL  
COMMANDS AND TYPE THEM TO SYNC UP.  
IF YOU GET ERRORS, SAVE YOUR WORK  
ELSEWHERE, DELETE THE PROJECT,  
AND DOWNLOAD A FRESH COPY.



# GitHub - Fork and pull request

---

# GitHub - Fork

The screenshot shows a GitHub repository page for 'CSE6000 / Fall2016'. The page includes a header with navigation links like 'Pull requests', 'Issues', and 'Gist'. Below the header, there's a main content area for the repository 'CSE6000 / Fall2016'. The repository has 9 commits, 1 branch, 0 releases, and 1 contributor. A 'Clone or download' button is visible. The commit history lists several changes made by 'HanByul Yang' and others. At the bottom, there's a 'Course Description' section with details about the course.

**Fork** 0

Class page for Embedded computing for scientific and industrial imaging applications — Edit

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

**HanByul Yang** added serials table

assignments added serials table 15 minutes ago

slides Updates slides 7 days ago

.gitignore added git ignore (VisualStudio.gitignore) from https://github.com/git... 4 days ago

README.md Update README.md 7 days ago

README.md

## CSE6000 Fall2016

Class page for Embedded computing for scientific and industrial imaging applications

- Course: CSE6000-01, Dept. of CSE, Yonsei University
- Instructor: 이은정 ([eunjunglee@yonsei.ac.kr](mailto:eunjunglee@yonsei.ac.kr)), 양한별 ([yhbyhb@yonsei.ac.kr](mailto:yhbyhb@yonsei.ac.kr))
- ASTC 516, Mon 16:00 ~ 18:50

Course Description

# GitHub - Fork result

The screenshot shows a GitHub repository page for 'yhbhyb/Fall2016'. The title bar indicates the URL is <https://github.com/yhbhyb/Fall2016>. The repository name 'yhbhyb / Fall2016' is displayed prominently at the top, with a red box highlighting the text 'forked from CSE6000/Fall2016'. Below the title, there are buttons for 'Code', 'Pull requests 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. To the right, there are buttons for 'Unwatch 1', 'Star 0', 'Fork 1', and a bell icon.

Class page for Embedded computing for scientific and industrial imaging applications — Edit

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

This branch is even with CSE6000:master.

**HanByul Yang added serials table** Latest commit e0a452d 19 minutes ago

**assignments** added serials table 19 minutes ago

**slides** Updates slides 7 days ago

**.gitignore** added git ignore (VisualStudio.gitignore) from https://github.com/git... 4 days ago

**README.md** Update README.md 7 days ago

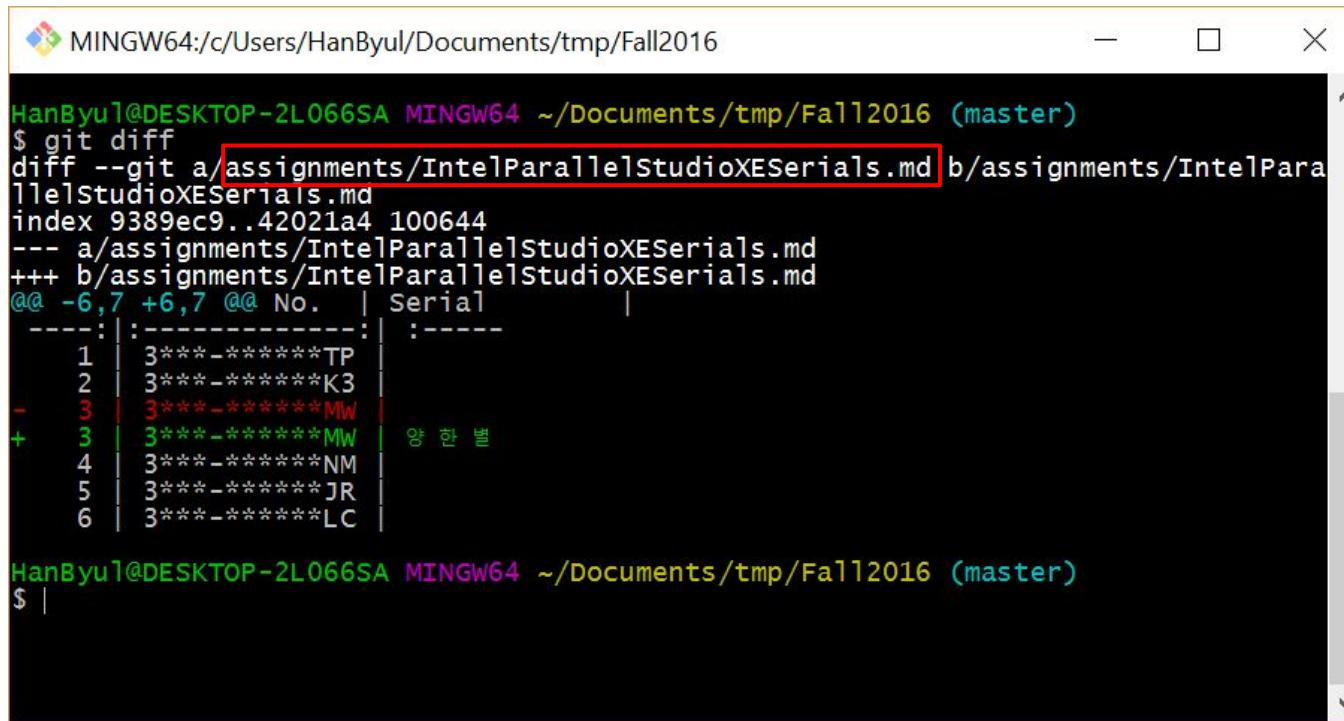
**README.md**

## CSE6000 Fall2016

Class page for Embedded computing for scientific and industrial imaging applications

- Course: CSE6000-01, Dept. of CSE, Yonsei University
- Instructor: 이은정 ([eunjunglee@yonsei.ac.kr](mailto:eunjunglee@yonsei.ac.kr)), 양한별 ([yhbhyb@yonsei.ac.kr](mailto:yhbhyb@yonsei.ac.kr))
- ASTC 516, Mon 16:00 ~ 18:50

# GitHub - clone and edit file



The screenshot shows a terminal window titled "MINGW64:/c/Users/HanByul/Documents/tmp/Fall2016". The command \$ git diff is run, comparing two files: assignments/IntelParallelStudioXESerials.md (a) and assignments/IntelParallelStudioXESerials.md (b). The output highlights a difference at line 3, where the serial number "MW" is added in the b file. The terminal also shows the user's name HanByul and the commit message "양 한 별".

```
HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/tmp/Fall2016 (master)
$ git diff
diff --git a/assignments/IntelParallelStudioXESerials.md b/assignments/IntelParallelStudioXESerials.md
index 9389ec9..42021a4 100644
--- a/assignments/IntelParallelStudioXESerials.md
+++ b/assignments/IntelParallelStudioXESerials.md
@@ -6,7 +6,7 @@ No. | Serial |
-----:|-----:|-----
 1 | 3***-*****TP |
 2 | 3***-*****K3 |
- 3 | 3***-*****MW |
+ 3 | 3***-*****MW | 양 한 별
 4 | 3***-*****NM |
 5 | 3***-*****JR |
 6 | 3***-*****LC |

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/tmp/Fall2016 (master)
$ |
```

# GitHub - commit and push

```
MINGW64:/c/Users/HanByul/Documents/tmp/Fall2016
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   assignments/IntelParallelStudioXESerials.md

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/tmp/Fall2016 (master)
$ git commit -m "양 한 별 's serial"
[master fd2698f] 양 한 별 's serial
 1 file changed, 1 insertion(+), 1 deletion(-)

HanByul@DESKTOP-2L066SA MINGW64 ~/Documents/tmp/Fall2016 (master)
$ git push origin master
Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 390 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/yhbyhb/Fall2016.git
  e0a452d..fd2698f  master -> master
```

# GitHub - pull request

The screenshot shows a GitHub repository page for 'yhbhyb/Fall2016'. The repository is described as 'Class page for Embedded computing for scientific and industrial imaging applications'. It has 10 commits, 1 branch, 0 releases, and 1 contributor (HanByul). A red box highlights the 'New pull request' button under the branch dropdown. Below the commit list, there is a section for 'README.md'.

Branch: master **New pull request**

This branch is 1 commit ahead of CSE6000:master.

**HanByul Yang** 양한별's serial

Latest commit `fd2698f` a minute ago

File	Commit Message	Time
assignments	양한별's serial	a minute ago
slides	Updates slides	7 days ago
.gitignore	added git ignore (VisualStudio.gitignore) from https://github.com/git...	4 days ago
README.md	Update README.md	7 days ago

**README.md**

## CSE6000 Fall2016

Class page for Embedded computing for scientific and industrial imaging applications

- Course: CSE6000-01, Dept. of CSE, Yonsei University
- Instructor: 이은정 ([eunjunglee@yonsei.ac.kr](mailto:eunjunglee@yonsei.ac.kr)), 양한별 ([yhbhyb@yonsei.ac.kr](mailto:yhbhyb@yonsei.ac.kr))

<https://github.com/github/gitignore> ASTC 516, Mon 16:00 ~ 18:50

# GitHub - pull request (reviewing changes)

The screenshot shows a GitHub browser interface comparing two branches: 'CSE6000/Fall2016' (base fork) and 'yhbyhb/Fall2016' (head fork). The 'master' branch is selected for both. The comparison shows 1 commit, 1 file changed, and 1 addition/deletion. A green button labeled 'Create pull request' is highlighted with a red box.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base fork: CSE6000/Fall2016 ▾ base: master ▾ ... head fork: yhbyhb/Fall2016 ▾ compare: master ▾

✓ Able to merge. These branches can be automatically merged.

Create pull request

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Sep 11, 2016

HanByul Yang 양한별's serial fd2698f

Showing 1 changed file with 1 addition and 1 deletion.

Unified Split

2 assignments/IntelParallelStudioXESerials.md

		@@ -6,7 +6,7 @@ No.   Serial	
6	6	----- :----- :-----	
7	7	1   3***-*****TP	
8	8	2   3***-*****K3	
9	-	3   3***-*****MW	
9	+	3   3***-*****MW   양한별	
10	+	----- :----- :-----	

# GitHub - create pull request

The screenshot shows a GitHub repository page for 'CSE6000 / Fall2016'. The 'Pull requests' tab is selected. A red box highlights the repository name 'CSE6000 / Fall2016' in the header. Below it, there are tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. On the right, there are buttons for 'Unwatch' (6), 'Star' (3), 'Fork' (1), and a bell icon.

**Open a pull request**

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base fork: CSE6000/Fall2016 ▾ base: master ▾ ... head fork: yhbyhb/Fall2016 ▾ compare: master ▾

✓ Able to merge. These branches can be automatically merged.

양한별's serial

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Allow edits from maintainers. Learn more

**Create pull request**

1 commit 1 file changed 0 commit comments 1 contributor

# GitHub - result of pull request

The screenshot shows a GitHub pull request page. At the top, the repository name is '양한별's serial by yhbyhb'. The URL is https://github.com/CSE6000/Fall2016/pull/1. The navigation bar includes Personal, Open source, Business, Explore, Pricing, Blog, Support, This repository, Search, Sign in, and Sign up.

The main header shows the repository 'CSE6000 / Fall2016', 6 watches, 3 stars, 1 fork, and 1 pull request. Below this, there are tabs for Code, Issues (0), Pull requests (1), Pulse, and Graphs. The active tab is Pull requests.

The pull request details are for '#1 양한별's serial'. It shows 'Open' status, merging '1 commit' from 'yhbyhb:master' into 'CSE6000:master'. The commit was made by 'yhbyhb' 33 minutes ago. There is no description provided.

On the right side, there are sections for Labels (None yet), Milestone (No milestone), Assignees (No one assigned), and Participants (1 participant, showing a profile picture).

At the bottom, there is a call-to-action button 'Sign up for free' and a link 'to join this conversation on GitHub. Already have an account? Sign in to comment'.

The footer includes links for GitHub, API, Training, Shop, Blog, and About, along with the GitHub logo.

# GitHub - maintainer's view of pull request

The screenshot shows a GitHub pull request page for the repository "CSE6000 / Fall2016". The pull request is titled "양한별's serial #1" and is from the user "yhybyhb". The pull request has 1 commit and 1 file changed. The base branch is "CSE6000:master" and the head branch is "yhybyhb:master". The pull request has been commented on by "yhybyhb" 37 minutes ago, stating "No description provided.". The status bar indicates "This branch has no conflicts with the base branch" and "Merging can be performed automatically." A red box highlights the "Merge pull request" button.

CSE6000 / Fall2016

Pull requests 1

양한별's serial #1

yhybyhb wants to merge 1 commit into CSE6000:master from yhybyhb:master

Conversation 0 Commits 1 Files changed 1

yhybyhb commented 37 minutes ago

No description provided.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

Write Preview

Leave a comment

Unsubscribe

Notifications

Lock conversation

# GitHub - result of merged pull request

The screenshot shows a GitHub pull request page for the repository "CSE6000 / Fall2016". The pull request is titled "양한별's serial #1" and has been merged by "yhybyhb" 14 seconds ago. The merge commit hash is "4ac892d". The pull request has 1 commit and 1 file changed. A comment from "yhybyhb" states "No description provided.". The pull request is labeled as "CSE6000 member". It has no milestones or assignees. There is one participant, "yhybyhb". Notifications indicate that the user is receiving notifications because they modified the open/close state. The page includes a comment section at the bottom.

양한별's serial #1

Merged yhybyhb merged 1 commit into cse6000:master from yhybyhb:master 14 seconds ago

Conversation 0 Commits 1 Files changed 1

yhybyhb commented 38 minutes ago  
No description provided.

CSE6000 member

Labels None yet

Milestone No milestone

Assignees No one—assign yourself

1 participant

Avoid bugs by automatically running your tests.  
Continuous integration can help catch bugs by running your tests automatically.  
Merge your code with confidence using one of our continuous integration providers.

Write Preview AA B i Leave a comment

Notifications Unsubscribe  
You're receiving notifications because you modified the open/close state.

Lock conversation

# Links

- [Git Cheat Sheet - GitHub Education](#)
- [GitHub Guides](#)
  - [Hello world](#)
  - [Forking Projects](#)
- GitHub markdown
  - [Mastering Markdown](#)
  - [Markdown Cheatsheet](#)