



## CSS427 SOFTWARE ENGINEERING, KMUTT CONFIGURATION MANAGEMENT LAB SHEET.

In this year's lab, we will be using GitHub instead of a CVS tool. Both systems can track changes to the files in your project and inform you of conflicts created when more than one developer make changes to the same file. Each developer works in an individual directory and then merges the work from each after the work is complete. CVS does not maintain multiple versions of source code files but keeps a single copy and records of all of the changes that are made. However, Git works by maintaining snapshots that are entire files, not just initial files + changes.

Follow the instructions below to complete this lab.

### Installation and preparation

1. Sign up for GitHub account at [www.github.com](http://www.github.com)
2. Download and install Git software for PC or Mac. GitHub won't work on your local computer if you don't install Git. Install Git for Windows, Mac or Linux. (see <http://git-scm.com/downloads>)
3. Run git commands on the Git Bash shell (terminal).

### My Example Workflow Steps

1. Create a blank repository in GitHub with no readme file. Print hub status.

Search GitHub

Explore Gist Blog Help

drsuthep +

Owner: drsuthep

Repository name: CMLab

Great repository names are short and memorable. Need inspiration? How about **ducking-spice**.

Description (optional)

☒ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

☒ Initialize this repository with a README  
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Create repository

2. Assume you already have 1 Readme.txt file, 2 C programs (Program #1 and Program #2), and 1 MS Word document in your local PC. Make a git repository on that PC's folder and then push it to the GitHub repository.

- I have my files waiting in d:/data/dropbox/cpe333/gitlab, so I go there first using 'cd' to change directory.

```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ cd /d/data/dropbox/cpe333/gitlab
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab
$
```

- I list the contents using 'ls':

```
MINGW32:/d/data/dropbox/cpe333/gitlab
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab
$ ls
Lottery.c  PM_Lab-2014.docx  PrintGrade.c  Readme.txt  bloodgroup.c
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab
$
```

- I then make this a .git repository using "git init". Then I set the user name and email for this repository to "drsuthep" and "suthepmail@gmail.com":

```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ git init
Initialized empty Git repository in d:/data/dropbox/cpe333/gitlab/.git/
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git config user.name "drsuthep"
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git config user.email "suthepmail@gmail.com"
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- I then add all 4 files to the git repository using add. And then commit. Then push it to GitHub. Upon adding they become staged.

```

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git status
On branch master

Initial commit

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

    Lottery.c
    PM_Lab-2014.docx
    PrintGrade.c
    Readme.txt
    bloodgroup.c

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

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git add *

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git status
On branch master

Initial commit

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

    new file:   Lottery.c
    new file:   PM_Lab-2014.docx
    new file:   PrintGrade.c
    new file:   Readme.txt
    new file:   bloodgroup.c

```

- Now we commit these changes with a note as "First copy of 5 files"

```

MINGW32:/d/data/dropbox/cpe333/gitlab
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git commit -m "First copy of 5 files"
[master (root-commit) 335fd7a] First copy of 5 files
5 files changed, 118 insertions(+)
create mode 100644 Lottery.c
create mode 100644 PM_Lab-2014.docx
create mode 100644 PrintGrade.c
create mode 100644 Readme.txt
create mode 100644 bloodgroup.c

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git status
On branch master
nothing to commit, working directory clean

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$

```

- The GitHub repository is located at <https://github.com/drsuthep/CMLab.git> as can be seen here:

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH https://github.com/drsuthep/CMLab.git

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

or create a new repository on the command line

- Add Collaborator as User B: "suthep64"

Unwatch 1 Star 0 Fork 0

- Options
- Collaborators**
- Webhooks & Services
- Deploy keys

**Collaborators** Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

suthep64 Add collaborator

- So I now link that remote repository to this repository.

```
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git remote add origin https://github.com/drsuthep/CMLab.git

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- Now I push my local folder (called origin master) to the GitHub repository.

```
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git push origin master
Username for 'https://github.com': drsuthep
Password for 'https://drsuthep@github.com':
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 105.50 KiB | 0 bytes/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/drsuthep/CMLab.git
 * [new branch]      master -> master

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- 3. User B creates a copy of that repository locally in another folder of his/her own computer. Print local status Users B.

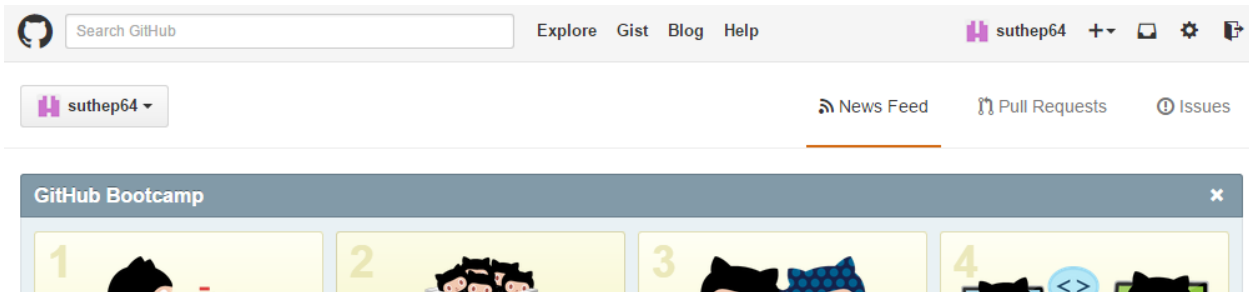
- Let's assume on user B's computer it is in folder d:/data/dropbox/cpe333/gitlab-userB:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

CPE-PC@CPE ~
$ cd /d/data/dropbox/cpe333/gitlab-userB

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB
$
```

- I will use the second user or user B as "suthep64" (I used 2 email accounts to simulate 2 users for myself). This account has email of "suthep@kmutt.ac.th", my other email account.



- I initialize the local repository at the current folder to create a .git repository and then set the user name and password configurations for this repository:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

$ git init
Initialized empty Git repository in d:/data/dropbox/cpe333/gitlab-userB/.git/

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git config user.name "suthep64"

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git config user.email "suthep@kmutt.ac.th"

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

- Now I inform the local directory where the source GitHub is and then pull that

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB
$ git remote add origin https://github.com/drsuthep/CMLab.git
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git pull origin master
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
From https://github.com/drsuthep/CMLab
 * branch                master      -> FETCH_HEAD
 * [new branch]          master      -> origin/master
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

4. User A change 1 function in Program #1. Print local status User A.
- Before Change file "bloodgroup.c":

```
File Edit Format View Help
#include <stdio.h>
#include <string.h>
void main()
{ char Name[40], G[5], answer[10];
  int i;
  do
  {
    printf("What is your Name? ");
    scanf("%s", Name);
```

- After Change file "bloodgroup.c":

```
File Edit Format View Help
#include <stdio.h>
#include <string.h>
void main()
{ char Name[50], G[5], answer[10];
  int i;
  do
  {
    printf("What is your name? ");
    scanf("%s", Name);
```

- User A's Git Status:

```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working
    modified:   bloodgroup.c

no changes added to commit (use "git add" and/or "git commit -a")
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- User B's Git Status:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git status
On branch master
nothing to commit, working directory clean
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

5. User B change 1 function in Program #2. Print local status User B.

- Before Change to "PrintGrade.c"

```
PrintGrade.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{
    char Name[30][40] = {"Sam Matthews", "Mary McBeth", "Ying Phakpon", "Chai Chanchai", "Yong
Boonsith", "Kitti Karndee"};
    int ID[30] = {5301, 5302, 5303, 5304, 5305, 5306};
    float GPA[30] = {2.73, 2.35, 3.65, 3.24, 1.32, 3.26};
    int N = 6, i;
```

- User B makes change to "PrintGrade.c":

```
PrintGrade.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{
    char Name[30][40] = {"James Jones", "Mary McBeth", "Ying Phakpon", "Chai Chanchai", "Yong
Boonsith", "Kitti Karndee"};
    int ID[30] = {5301, 5302, 5303, 5304, 5305, 5306};
    float GPA[30] = {3.73, 2.35, 3.65, 3.24, 1.32, 3.26};
    int N = 6, i;
```

- User B Status:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

$ git status
On branch master
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:   PrintGrade.c

no changes added to commit (use "git add" and/or "git commit -a")
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

6. User A create a new C program Program #3. Do not put into Git yet. Print local status User A.
- A new file "TimeGreeting.c" was added to user A's folder.

```
MINGW32:/d/data/dropbox/cpe333/gitlab

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ ls
Lottery.c          PrintGrade.c  TimeGreeting.c
PM_Lab-2014.docx  Readme.txt   bloodgroup.c

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- The new Status:

```
MINGW32:/d/data/dropbox/cpe333/gitlab

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git status
On branch master
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:   bloodgroup.c

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

        TimeGreeting.c

no changes added to commit (use "git add" and/or "git commit -a")
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

7. User A adds Program #3 to Git control (git add). Print local status User A.  
**\$ git add TimeGreeting.c** //Please add this missing line.



```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   TimeGreeting.c

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working)

    modified:   bloodgroup.c

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

8. User A commits changes (note that changed are Program #1 and Program #3). Print local status User A.
- I commit with the statement "Add 1 file. Change 1 file"

```
MINGW32:/d/data/dropbox/cpe333/gitlab
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git commit -m "Add 1 file. Change 1 file"
[master ef7d22e] Add 1 file. Change 1 file
1 file changed, 49 insertions(+)
create mode 100644 TimeGreeting.c

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- With git status I notice I forgot to add and commit 1 file. You can add and commit at once by using "git commit -a" or otherwise you do 2 steps: 1) git add bloodgroup.c 2) git commit -m "Modified bloodgroup.c".

```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ git status
On branch master
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:   bloodgroup.c

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

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
```

```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ git commit -a -m "Modified bloodgroup.c"
[master 7546e91] Modified bloodgroup.c
1 file changed, 2 insertions(+), 2 deletions(-)

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git status
On branch master
nothing to commit, working directory clean

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

9. User A pushes changes to the GitHub repository. Print hub status. Print User A status.

```
MINGW32:/d/data/dropbox/cpe333/gitlab

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git push origin master
Username for 'https://github.com': drsuthep
Password for 'https://drsuthep@github.com':
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.14 KiB | 0 bytes/s, done.
Total 6 (delta 3), reused 0 (delta 0)
To https://github.com/drsuthep/CMLab.git
335fd7a..7546e91 master -> master

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

10. User B stages its changes (Program #2 had changed) and commits at once. Print local status User B.

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git status
On branch master
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:   PrintGrade.c

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

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git commit -a -m "Modified PrintGrade.c"
[master 0446152] Modified PrintGrade.c
1 file changed, 2 insertions(+), 2 deletions(-)

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

11. User B pulls repository from GitHub.

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git pull origin master
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 3), reused 6 (delta 3), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/drsuthep/CMLab
* branch      master      -> FETCH_HEAD
335fd7a..7546e91 master    -> origin/master
Merge made by the 'recursive' strategy.
TimeGreeting.c | 49 ++++++
bloodgroup.c   | 4 ++--
2 files changed, 51 insertions(+), 2 deletions(-)
create mode 100644 TimeGreeting.c

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

12. User B pushes changes to GitHub repository. Print status User B.

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git push origin master
Username for 'https://github.com': suthep64
Password for 'https://suthep64@github.com':
Counting objects: 8, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 574 bytes | 0 bytes/s, done.
Total 5 (delta 3), reused 0 (delta 0)
To https://github.com/drsuthep/CMLab.git
7546e91..c0f9462 master -> master

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

13. User A pulls repository from GitHub.

```
MINGW32:/d/data/dropbox/cpe333/gitlab
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git status
On branch master
nothing to commit, working directory clean

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git pull origin master
remote: Counting objects: 5, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 3), reused 5 (delta 3), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/drsuthep/CMLab
* branch      master      -> FETCH_HEAD
7546e91..c0f9462 master    -> origin/master
Updating 7546e91..c0f9462
Fast-forward
PrintGrade.c | 4 ++--
1 file changed, 2 insertions(+), 2 deletions(-)

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- All 3 repositories have same copy now.

14. Then User A changes Program #1 and User B change Program #1, both at a different location. Both add and commit changes in one statement. Print status of User A and User B.

- Original Lottery.c file:

```
Lottery.c - Notepad
File Edit Format View Help
void main()
{
    int WinLotteryNo[10] = {1522, 1711, 5515, 7233, 5614, 8876, :
    float WinAmount[10] = {50000, 10000, 10000, 5000, 5000, 5000.
    int N = 10;
    int i, won;
```

- Change by User A:

```
Lottery.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{
    int WinLotteryNo[10] = {1522, 1711, 5515, 7233, 5614, 8876, 1
    float WinAmount[10] = {50000, 15000, 10000, 5000, 5000, 5000,
    int N = 10;
    int i, won; |
```

- Change by User B:

```
Lottery.c - Notepad
File Edit Format View Help
void main()
{
    int WinLotteryNo[10] = {1579, 1711, 5515, 7233, 5614, 8876, 1
    float WinAmount[10] = {50000, 10000, 10000, 5000, 5000, 5000,
    int N = 10;
    int i, won;
```

- Commit by User A:

```
MINGW32:/d/data/dropbox/cpe333/gitlab
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git commit -a -m "lottery change win 2"
[master 0fe27d8] lottery change win 2
1 file changed, 1 insertion(+), 1 deletion(-)
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- Commit by User B:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB
$ git commit -a -m "change first prize lottery no."
[master 0038a31] change first prize lottery no.
1 file changed, 1 insertion(+), 1 deletion(-)

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

15. They both push changes to GitHub Repository.

- User A:

```
MINGW32:/d/data/dropbox/cpe333/gitlab

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$ git push origin master
Username for 'https://github.com': drsuthep
Password for 'https://drsuthep@github.com':
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To https://github.com/drsuthep/CMLab.git
c0f9462..0fe27d8 master -> master

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- User B cannot push because some changes were there, so got to pull first. It's always a good idea to pull and then push.

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git push origin master
Username for 'https://github.com': suthep64
Password for 'https://suthep64@github.com':
To https://github.com/drsuthep/CMLab.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/drsuthep/CMLab.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

- So Pull first and we realize a merge conflict:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 2), reused 3 (delta 2), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/drsuthep/CMLab
* branch          master      -> FETCH_HEAD
  c0f9462..0fe27d8  master      -> origin/master
Auto-merging Lottery.c
CONFLICT (content): Merge conflict in Lottery.c
Automatic merge failed; fix conflicts and then commit the result.

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master|MERGING)
$
```

- Due to conflict the 2 changes in the files are merged in Lottery.c. We have to resolve it manually:

```
Lottery.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{
<<<<<< HEAD
    int WinLotteryNo[10] = {1579, 1711, 5515, 7233, 5614, 8876, 1215, 4232, 9155, 9995};
    float WinAmount[10] = {50000, 10000, 10000, 5000, 5000, 5000, 5000, 3000, 3000, 3000};
    =====
    int WinLotteryNo[10] = {1522, 1711, 5515, 7233, 5614, 8876, 1215, 4232, 9155, 9995};
    float WinAmount[10] = {50000, 15000, 10000, 5000, 5000, 5000, 5000, 3000, 3000, 3000};
    >>>>>> 0fe27d84b23bb0224371b3b60020f91e54cd0f4d
    int N = 10;
    int i, won;
```

- We fix it to the following and then save the file in user B:

```
Lottery.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{

    int WinLotteryNo[10] = {1579, 1711, 5515, 7233, 5614, 8876, 1215, 4232, 9155, 9995};
    float WinAmount[10] = {50000, 15000, 10000, 5000, 5000, 5000, 5000, 3000, 3000, 3000};
    int N = 10;
    int i, won;
```

- We then commit the change:

```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master|MERGING)
$ git commit -a -m "Corrected Merge Conflict of Lottery.c"
[master 9a2d945] Corrected Merge Conflict of Lottery.c

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

- We then push the changes:



```
MINGW32:/d/data/dropbox/cpe333/gitlab-userB
CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$ git push origin master
Username for 'https://github.com': suthep64
Password for 'https://suthep64@github.com':
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 636 bytes | 0 bytes/s, done.
Total 6 (delta 4), reused 0 (delta 0)
To https://github.com/drsuthep/CMLab.git
   0fe27d8..9a2d945  master -> master

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab-userB (master)
$
```

16. Then User A pulls the changes.

```
MINGW32:/d/data/dropbox/cpe333/gitlab
$ git pull origin master
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 6 (delta 4), reused 6 (delta 4), pack-reused 0
Unpacking objects: 100% (6/6), done.
From https://github.com/drsuthep/CMLab
 * branch                master      -> FETCH_HEAD
   0fe27d8..9a2d945  master      -> origin/master
Updating 0fe27d8..9a2d945
Fast-forward
 Lottery.c | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

CPE-PC@CPE /d/data/dropbox/cpe333/gitlab (master)
$
```

- Now user A has the same copy of the file Lottery.c with conflict resolved as User B:

```
Lottery.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{
    USER A

    int WinLotteryNo[10] = {1579, 1711, 5515, 7233, 5614, 8876, 1215, 4232, 9155, 9995};
    float WinAmount[10] = {50000, 15000, 10000, 5000, 5000, 5000, 5000, 3000, 3000, 3000};
    int N = 10;
}

Lottery.c - Notepad
File Edit Format View Help
#include <stdio.h>
void main()
{
    USER B

    int WinLotteryNo[10] = {1579, 1711, 5515, 7233, 5614, 8876, 1215, 4232, 9155, 9995};
    float WinAmount[10] = {50000, 15000, 10000, 5000, 5000, 5000, 5000, 3000, 3000, 3000};
    int N = 10;
}
```

## Now Your Turn

Run the following and submit a PDF dump similar to my above workflow dump for the following commands:

1. Create a blank repository in GitHub under User A's account called "GITLAB" with 1 Readme file. Print hub

- status. Both users A and B will use this repository. Add User B as collaborator to this repository.
2. In some path create a folder called “CPE333-A” for User A. Make this a git repository and pull data from GitHub. Show status.
  3. In some path create a folder called “CPE333-B” for User B. Make this a git repository and pull data from GitHub. Show status.
  4. User A. Use file manager to put 2 C programs (Program #1 and Program #2), and 1 MS Word document into CPE333-A.
  5. User A. Add all files to git. Then commit. Show Status.
  6. User A. Push your local repository to the GitHub repository. (You should always pull before you push). Show screen.
  7. User B pulls data from GitHub. Show status.
  8. User B change 1 function in Program #2. Print local status User B.
  9. User B create a new C program Program #3. Do not put into Git yet. Print local status User B.
  10. User B adds Program #3 to Git control (git add). Print local status User B.
  11. User B commits changes. Print local status User B.
  12. User B pushes changes to the GitHub repository. (You should always pull before you push). Print hub status. Print User B status.
  13. User A change something in C Program #1. Print local status User A.
  14. User A stages its changes and commits at once. Print local status User A.
  15. User A pulls repository from GitHub.
  16. User A pushes changes to GitHub repository. Print status User A.
  17. User B pulls repository from GitHub.
    - All 3 repositories have same copy now.
  18. Then User A changes Program #1 at 2 places and User B changes same Program #1 at 2 places (make sure they change same general error or in the next step there may be no merge conflict visible ☺). Both add and commit changes. Print status of User A and User B.
  19. User A pulls from GitHub and then pushes to GitHub. User B then pulls and pushes to GitHub, but there's conflict. (There will be a conflict if user A and B both change Program #1 at about the same lines). Resolve this merge conflict for User B and commit change. Show each step.
  20. Then User A pulls the changes. Show the Program#1 for both users that they are now the same.