

GitHub Intro Day 2

Using GitHub in an IDE of your choice + Merge Conflicts

GitHub Intro Day 2 Goals

- I will show GitHub on Xcode for Swift (iPhone & iPad & Mac development software)
- IntelliJ
- Visual Studio Code
- Dealing with Merge Conflicts
- Any IDE requests?

Quick Announcement



Computer Science

**FOLLOW US ON
SOCIAL MEDIA**

LinkedIn



Instagram



VCU College of
Engineering

Important Note! Before we Begin

- Sometimes, even when you're signed in to GitHub on your IDE it may still say you need an email and username for Git
- Run the commands on the right
 - Use your GitHub or VCU associated email
 - And anything for your name
- This is used in our commit signatures

```
*** Please tell me who you are.  
Run  
  
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"  
  
to set your account's default identity.  
Omit --global to set the identity only in this repository.
```

Visual Studio Code (VSCode)

- Click the weird wiggly line on the left
- Then click Publish to GitHub
 - You will be prompted to log in through your web browser

```
serial.cpp mpi.cpp Makefile
cuda.cu printStuff(float *, float *, int, int)
void *threadCode(void * ptr){
    int max_value = -1;
    int max_class = 0;
    for(int i = 0; i < num_
        if(classCounts[i] >
            max_value = classCounts[i];
            max_class = i;
        }
    }
    // Make prediction with
    predictions[queryIndex];
    // Reset arrays
    for(int i = 0; i < 2*k;
        memset(classCounts, 0,
    }
    //pthread_exit(0);
}

__global__ void printStuff(float *
int idx = blockIdx.x * bloc
```

The folder currently open doesn't have a Git repository. You can initialize a repository which will enable source control features powered by Git.

Initialize Repository

To learn more about how to use Git and source control in VS Code [read our docs](#).

You can directly publish this folder to a GitHub repository. Once published, you'll have access to source control features powered by git and GitHub.

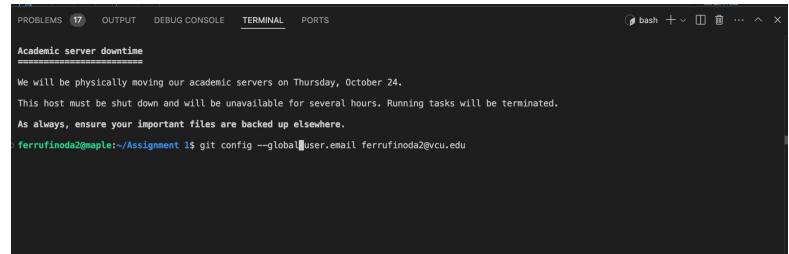
Publish to GitHub

The extension 'GitHub' wants to sign in using GitHub.

Cancel Allow

Visual Studio (VSCode)

- Before you begin to set up GitHub, we must set up Git
- We will give Git our e-mail and a username
 - Make the email the same as your VCU email or GitHub account email
 - The user name can be anything
- To do this, go into the terminal and run the commands in the screenshot



PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS

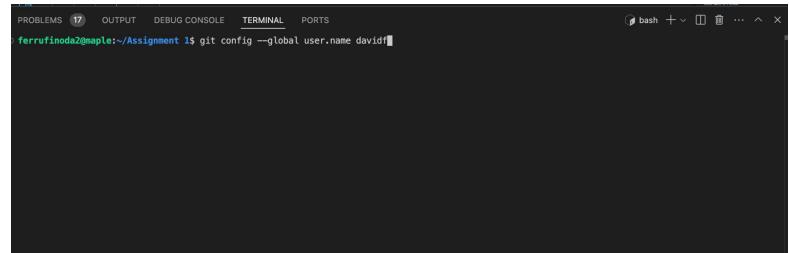
Academic server downtime

We will be physically moving our academic servers on Thursday, October 24.

This host must be shut down and will be unavailable for several hours. Running tasks will be terminated.

As always, ensure your important files are backed up elsewhere.

```
ferrufinoda2@maple:~/Assignment $ git config --global user.email ferrufinoda2@vcu.edu
```

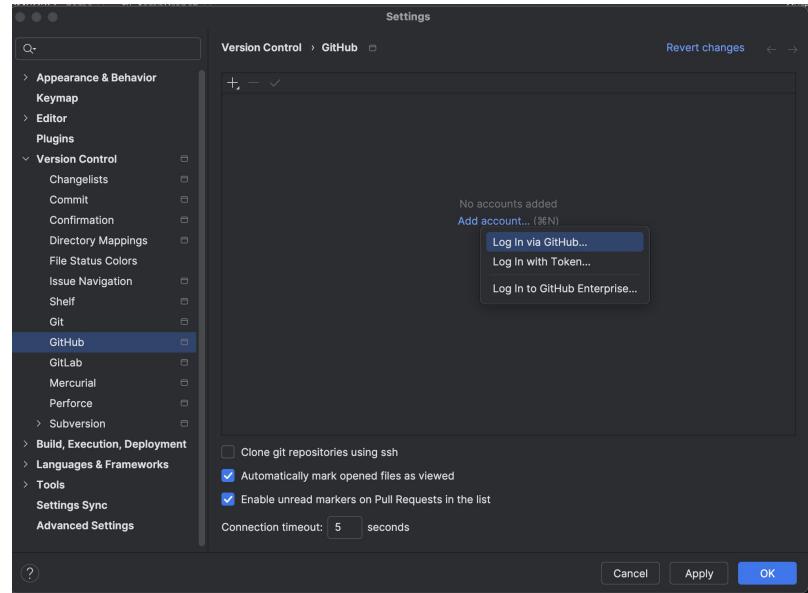


PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
ferrufinoda2@maple:~/Assignment $ git config --global user.name david
```

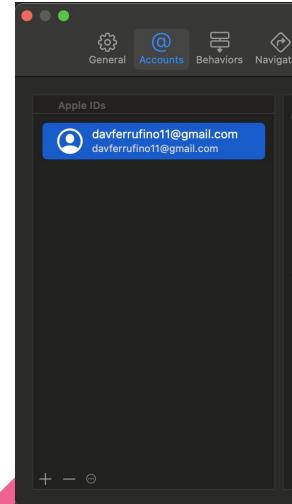
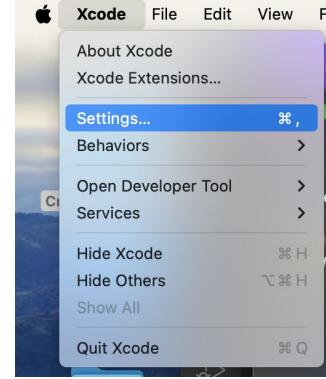
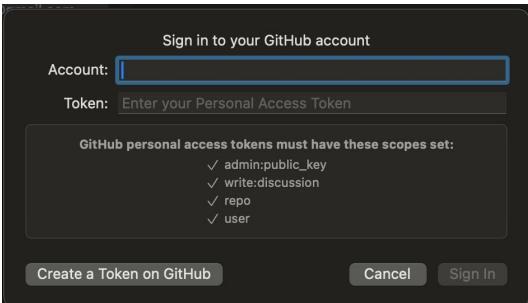
IntelliJ <3

- File → Settings → Version Control → GitHub
 - Add your Account and Log In Via GitHub



XCode

- Settings → Accounts
 - Then click the little plus in the bottom
 - Click GitHub and you should see a screen that looks like this
- Click Create a Token on GitHub



What is a token?

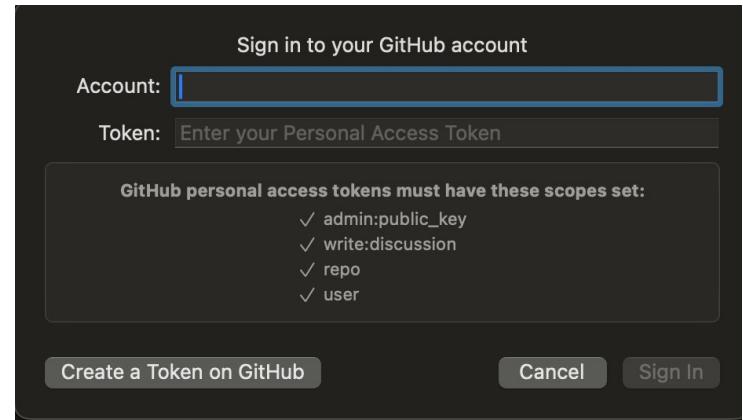
- A token is a unique identifier that allows access to your account
- If you have the token = you have access to the account
- But you can limit how much access each token has
- You can also set them to automatically expire after a certain time

The screenshot shows the GitHub 'Personal access tokens' section. At the top, there are links for 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens'. Below these are 'Fine-grained tokens' and 'Tokens (classic)', with 'Tokens (classic)' being highlighted. A 'Beta' button is visible. On the right, a large heading says 'New personal access token (classic)'. Below it, a note states: 'Personal access tokens (classic) function like ordinary OAuth access tokens. They can be for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#)'. There is a 'Note' input field with a placeholder 'What's this token for?'. An 'Expiration' dropdown is set to '30 days', with a note below stating 'The token will expire on Fri, Nov 8 2024'. Under 'Select scopes', it says 'Scopes define the access for personal tokens. [Read more about OAuth scopes](#)'. A table lists scopes with checkboxes:

<input type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events

Let's make our XCode token

- If you noticed, we needed certain permissions to make this token
 - Admin:public_key
 - Write:discussion
 - Repo
 - User
- When we are creating our token, we must make sure that it includes these permissions



Let's make sure our token has the right permissions

- On the token creation page, check these scopes
 - Then scroll down and hit generate token

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

What's this token for?

Expiration *

The token will expire on Fri, Nov 8 2024

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input type="checkbox"/> write:repo_hook	Write repository hooks
<input type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input checked="" type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories
<input checked="" type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions

Your token is now generated!

- Copy it right now!!!!
 - Or it will be gone forever and you'll have to delete and remake
- And paste it into the token section in Xcode, along with your username in the name section
 - And just hit Sign In

Personal access tokens (classic)

[Generate new token ▾](#)

Tokens you have generated that can be used to access the [GitHub API](#).

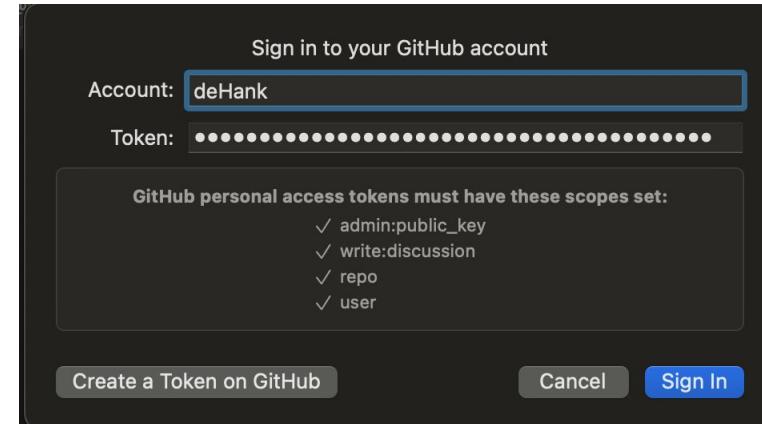
⚠ Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_W9yTjYkPGwdmRqvqkRYqpgrGsuudoU4X0vF1 [Copy](#) [Delete](#)

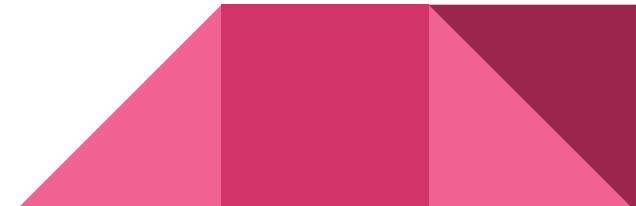
macos-xcode-updated — admin:public_key, repo, user, write:discussion
Last used within the last 2 months
Expires on Mon, Nov 4 2024.

MacOS Xcode — project, repo
Last used within the last 3 weeks
Expires on Mon, Oct 28 2024.

iOS Client vcu mac — admin:org, admin:org_hook, admin:public_key, admin:repo_hook, delete_repo, repo, user, write:discussion, write:packages
Last used within the last week
Expires on Sun, Aug 31 2025.



Any questions?



Merge Conflicts

- Q from last class: What do I do if I get a merge conflict? I.e. Two or more group members write to the same file, and one gets pushed to main first
 - Now the second group member to push to main will be outdated!

Merge Conflicts continued!

- In this example, I will create a conflict:
- One person makes changes, and we merge those changes into main
- Notice how both of these branches are overwriting the same parts of the same file
 - Once one is added to main, the other will conflict (shown on next slide)

added hi CMSC 355! #3

A screenshot of a GitHub pull request interface. The title is "added hi CMSC 355! #3". The commit message is "deHank wants to merge 1 commit into `main` from `adding-CMSC-355`". The commit details show a single file change: `index.html`. The diff shows the following code:

```
@@ -8,5 +8,6 @@
 8   8   </head>
 9   9   <body>
10  10     <h1>Hello, VCU!</h1>
11  11 +   <p>Hi CMSC 355!</p>
12  12   </body>
13  13   </html>
```

The line `<p>Hi CMSC 355!</p>` is highlighted in green, indicating it was added by the branch being merged.

causing a conflict! #4

A screenshot of a GitHub pull request interface. The title is "causing a conflict! #4". The commit message is "deHank wants to merge 2 commits into `main` from `causeConflict`". The commit details show a single file change: `index.html`. The diff shows the following code:

```
@@ -8,5 +8,6 @@
 8   8   </head>
 9   9   <body>
10  10     <h1>Hello, RVA!</h1>
11  11 +   <p>This will cause a merge conflict!</p>
12  12   </body>
13  13   </html>
```

The line `<p>This will cause a merge conflict!</p>` is highlighted in green, indicating it was added by the branch being merged.

Merging one branch, causes a conflict with the other

 Commit message

Merge pull request #3 from deHank/adding-CMSC-355

Extended description

added hi CMSC 355!

Commit email

ferrufinoda2@vcu.edu

Confirm merge **Cancel**

causing a conflict! #4

 Open deHank wants to merge 2 commits into main from causeConflict

Conversation 0 Commits 2 Checks 0 Files changed 1

deHank commented 5 minutes ago
No description provided.

deHank added 2 commits 8 minutes ago

causing a conflict!

Merge branch 'main' of github.com:deHank/CMSC35500 into causeConflict

This branch has not been deployed
No deployments

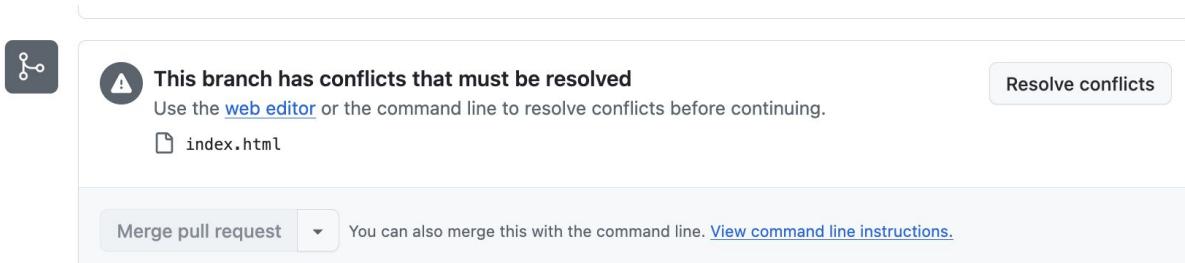
This branch has conflicts that must be resolved
Use the [web editor](#) or the command line to resolve conflicts before continuing.
index.html

Merge pull request You can also merge this with the command line. [View command line instructions.](#)

Add a comment

To resolve:

- A couple of options, use GitHub's built in conflict resolver



- Or, you can use your IDE's conflict resolver and pull from origin

Github Editor

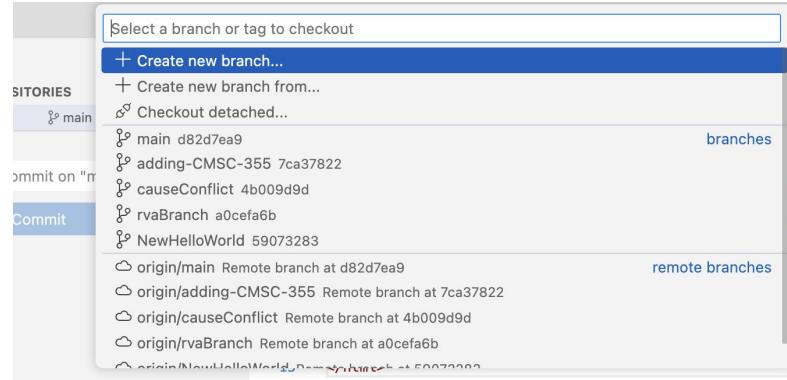
- The text between <<<< and ===== will be from one branch
- The and the text between === and >>>> will be from another
- You pick and delete manually
- Downside of the GitHub editor?
 - No compiling!!! You better be a machine at coding!

index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hello World</title>
8  </head>
9  <body>
10     <h1>Hello, RVA!</h1>
11     <<<<< causeConflict
12     <p>This will cause a merge conflict!</p>
13     =====
14     <p>Hi CMSC 355!</p>
15     >>>>> main
16  </body>
17  </html>
```

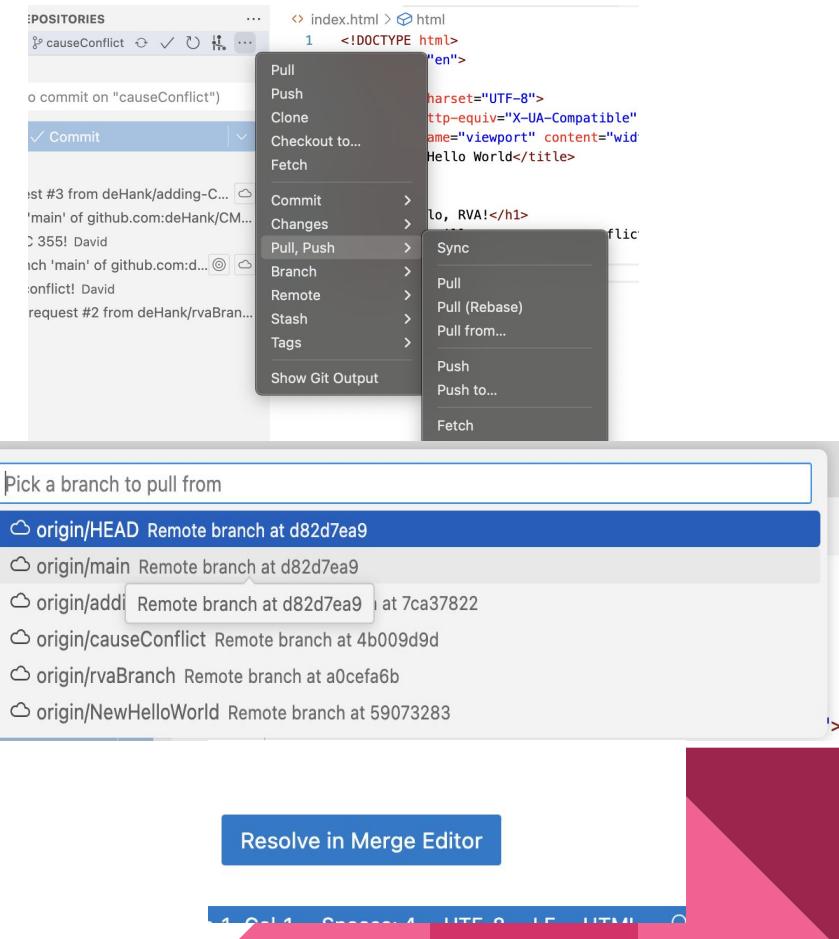
Instead! Use your IDE!

- Many IDE's support merging with a GUI
- IDE's handle it in many different ways, but today I'll show you VSCode
- Step 1: Switch to the branch CAUSING the conflict (the one that is second to merge into main)



Instead! Use your IDE (Cont...)

- Step 2: Click pull, then pull from
- Step 3: Click pull from **ORIGIN MAIN**
- **A resolve in merge editor button will pop up in the bottom right**
 - (you may have to x an error box that is covering it)



IDE to fix merge conflict (cont...)

- You will be able to pick and choose, if you want to let the IDE
 - You can allow the IDE to try and combine the changes, or you can combine them manually

The screenshot shows a merge conflict in an `index.html` file. The left pane shows the incoming version (d82d7ea) and the right pane shows the current version (4b009d9). The conflict occurs at line 11, where both sides have added a `<h1>Hello, RVA!</h1>` tag. A yellow box highlights the conflict area, containing the message "Accept Incoming | Accept Current | Ignore". The bottom pane shows the resulting merged code, which contains both `<h1>Hello, RVA!</h1>` tags. A status bar at the bottom right indicates "1 Conflict Remaining".

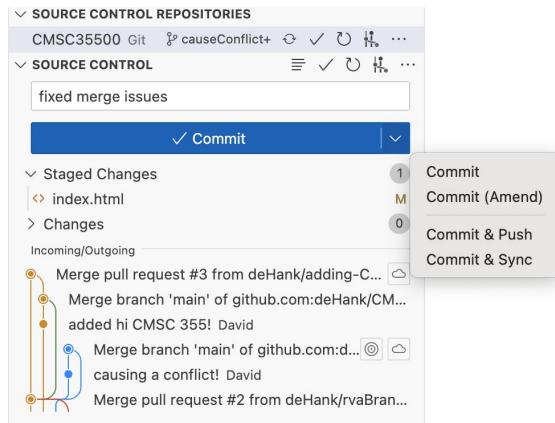
```
<html lang="en">
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello, RVA!</h1>
<p>HL OMSC 355!</p>
</body>
</html>
```

```
<html lang="en">
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello, RVA!</h1>
<h1>Hello, RVA!</h1>
<p>This will cause a merge conflict!</p>
</body>
</html>
```

```
<html lang="en">
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello, RVA!</h1>
<h1>Hello, RVA!</h1>
No Changes Accepted
</body>
</html>
```

IDE to fix merge conflict (cont...)

- Then hit complete merge, and you can push to your branch
- The pull request you can make, will now be able to automatically merge!



The screenshot shows the code editor with two tabs: 'Incoming' and 'Current'. Both tabs show the same content for 'index.html':

```
<html lang="en">
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello, RVA!</h1>
<p>Hi CMSC 355!</p>
</body>
</html>
```

The 'Result' pane at the bottom shows the merged code:

```
<html lang="en">
<head>
<title>Hello World</title>
</head>
<body>
<h1>Hello, RVA!</h1>
<p>This will cause a merge conflict!</p>
<p>Hi CMSC 355!</p>
</body>
</html>
```

A 'Complete Merge' button is visible in the bottom right corner of the result pane.

Success!!!



No conflicts with base branch

Merging can be performed automatically.

Merge pull request



You can also merge this with the command line. [View command line instructions.](#)

To add collaborators

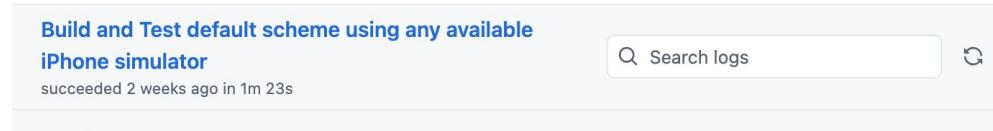
- In your repo go to settings → collaborators
- Then add people and add your groupmates

The screenshot shows the GitHub repository settings page for 'deHank / 603assignment02'. The 'Settings' tab is active. On the left, a sidebar lists various settings categories: General, Access (with 'Collaborators' selected), Code and automation, Security, and Integrations. The 'Access' section contains sub-links for Branches, Tags, Rules, Actions, Webhooks, Codespaces, and Pages. The 'Who has access' section indicates it's a 'Private repository'. It shows two sections: 'PRIVATE REPOSITORY' (Only those with access to this repository can view it) and 'DIRECT ACCESS' (0 collaborators have access to this repository. Only you can contribute to this repository). Below this is the 'Manage access' section, which displays a message: 'You haven't invited any collaborators yet' and a green 'Add people' button.

More Things you can do with GitHub!!

- You can Lint and Prettier code!
 - What the heck is this? It's a standardized formatting
 - Useful for when multiple people with different styles contribute to the same project
- You can also run code tests to ensure groupmates' changes actually work!
 - Find out who broke the codebase in real time!

```
diff --git Neuro App/KeyChainHelper.swift Neuro App/KeyChainHelper.swift
index 26,10 +26,9 @@ struct KeychainHelper {
@@ -26,10 +26,9 @@ struct KeychainHelper {
 26 26         kSecClass as String: kSecClassInternetPassword,
 27 27         kSecAttrAccount as String: account
 28 28     ]
 29 29     -
 30 30     -
 31 30     let attributesToUpdate: [String: Any] = [
 32 32     -         kSecValueData as String: passwordData,
 33 31     +         kSecValueData as String: passwordData
 34 32     ]
 35 34     // Attempt to update the existing item.
```



Questions?

