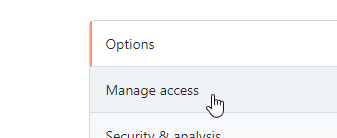
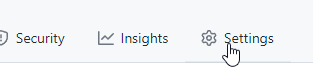
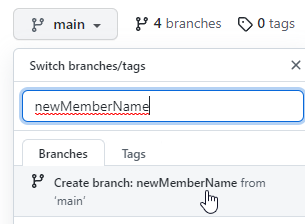
**GITHUB COLLAB 101**

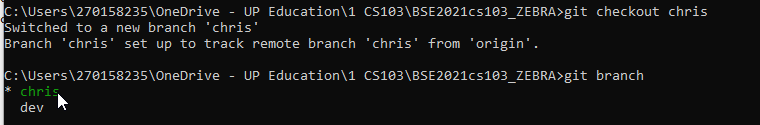
**GITMASTER QUICKSTART**  
(Gitmaster = person who creates the initial repo on Github)

**On Github**

1. Create a repository on Github
2. Click on Settings > Manage Access  
     
   
3. Add a hooman bean as a collaborator
4. Add branches for dev, teamMember1, and teamMember2   
   

**On your local machine**

On your local machine command prompt:

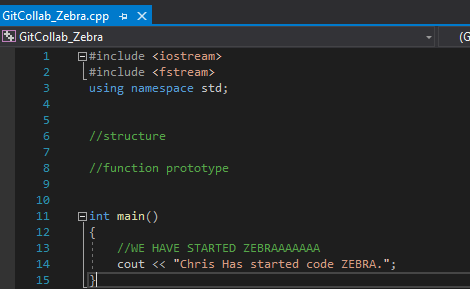
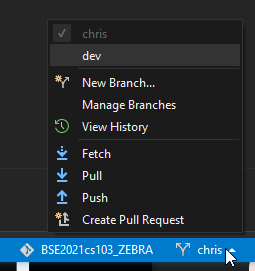
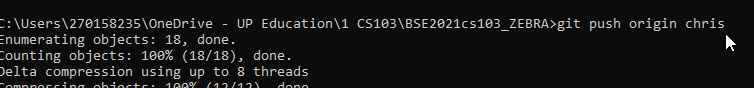
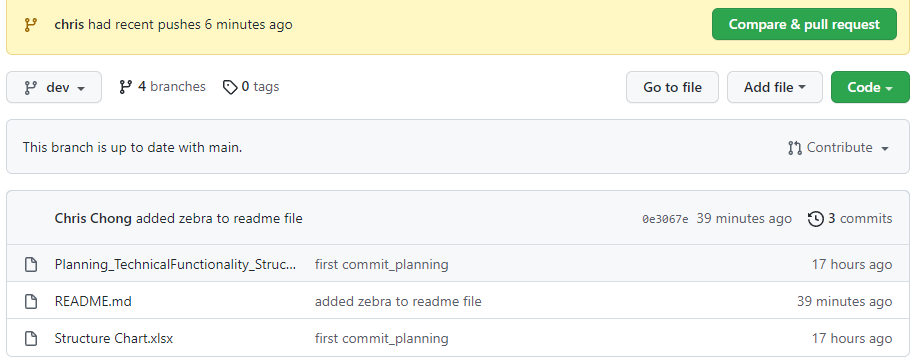
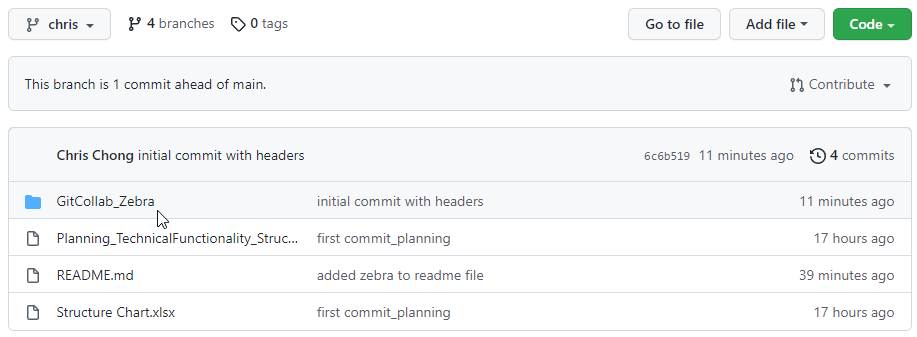
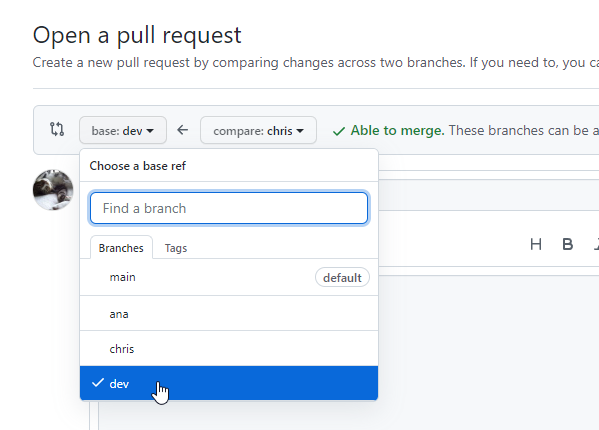
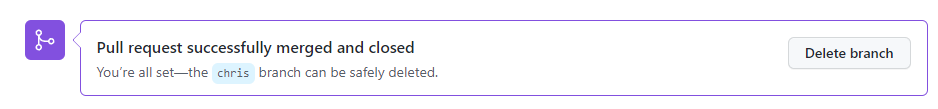
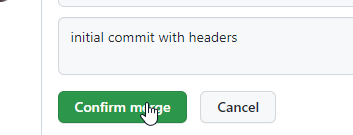
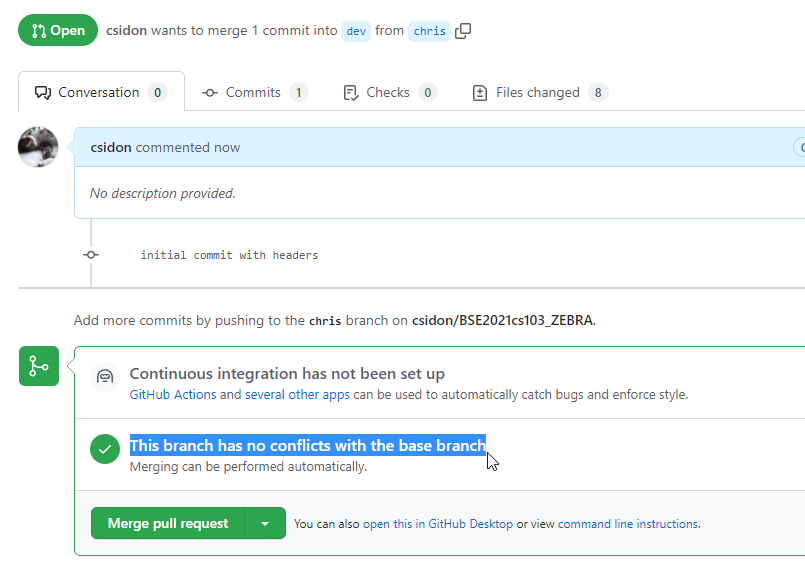
1. Cd into the folder where you would like to clone the repo.
2. Get the repository URL from Github, then git clone the dev branch using   
   **git clone <<URL>> -b dev**(-b dev makes sure that you’re pulling from the branch “dev”)
3. Check which branch you’re in using git branch  
   ^ This shows that you’re now working in the dev branch.
4. To work in your branch, use **git checkout <<branchName>>**(If you check again using **git branch**, it should show that you’re now working in your branch)

# **Merging changes with Github**

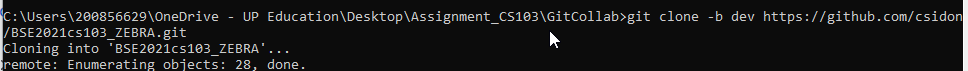
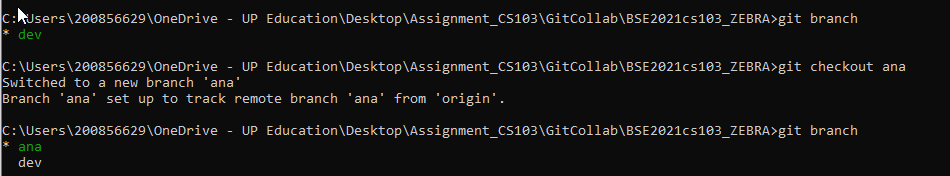
In this section, we are going to:

1. Make some changes (this example uses Visual Studio)
2. Make sure that you are working in the right branch in VS (if you’re using VS)
3. Push the changes to YOUR BRANCH
4. In Github, merge the changes to the DEV BRANCH

Remember, always **PULL from the dev branch,** and  
**PUSH to YOUR branch**, then **merge** it in Github!

1. Create a Visual Studio file and create a main structure  
    **!IMPORTANT Make sure you’re in your branch in Visual Studio! (Bottom right corner)**
2. Save your VS file, then do the usual:
   * **git add .**
   * **git commit -m “<<commit file comments>>”**
3. Now push it **to your own branch** using:  
     
   **git push origin <<branchName>>**
4. Go back to Github and run a quick check – You should see the file appear in your branch, but not in teamMember2’s branch or dev
5. Merge **your branch 🡪 dev branch**  
   (Note: If you pushed to your branch, *you* will need to do the merge. Your other team member will not be able to do that for you)  
     
   **!Important Check to make sure that you are merging from *your branch* to the dev branch!** 
6. Create the pull request and make sure that your branch has no conflicts with the base branch, then confirm merge. If your pull request turns a pretty **PURPLE**, you have succeeded in life. 

# **Now for Team Member 2**

1. Create a folder in your local machine and cd into it
2. Then clone the **dev branch** using   
     
   **git clone -b dev <<github url >>**
3. Cd into your newly created folder, then check which branch you’re working in using   
   **git branch**You should be in a dev branch, with no other branches. 
4. To get into your branch, use **git checkout <<branchName>>**Check which branch you’re working in (using git branch) again – you should now be working in your branch, indicated by an \* and green text  
   

Tada! Now refer to pages 3 – 6 (Merging changes with Github) to find out how to merge!

# **Resolving conflicts in Github**

We will be having conflicts. It will be painful. This is part of life.

Rule #1

KEEP BACKUPS. It will save you a (some) pain. And tears.