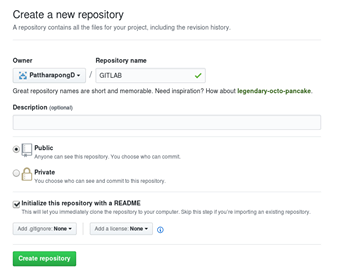
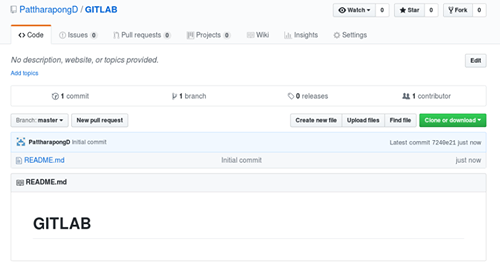
Now Your Turn

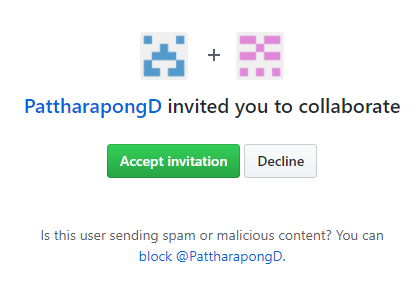
Run the following and submit a PDF dump similar to my above workflow dump for the following commands:

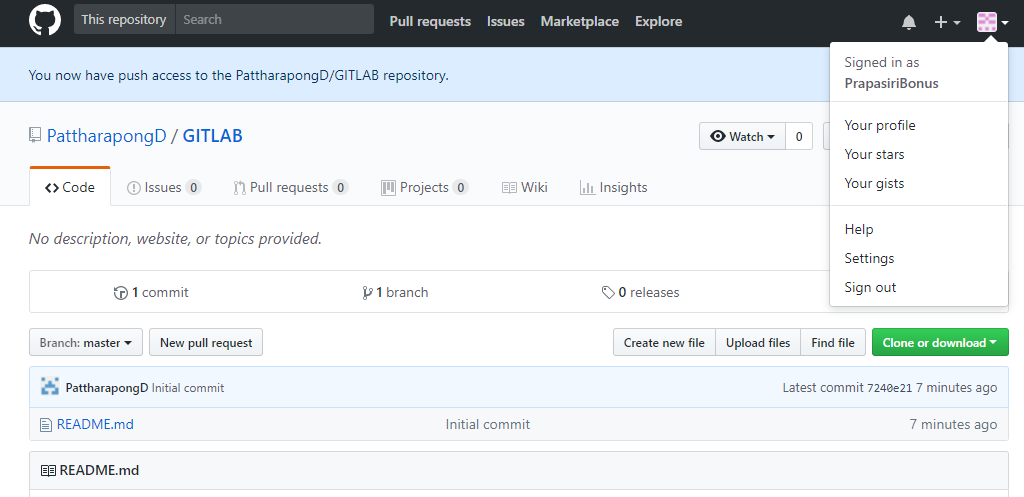
1. Create a blank repository in GitHub under User A’s account called “GITLAB” with 1 Readme file. Print hub

status. Both users A and B will use this repository. Add User B as collaborator to this repository.



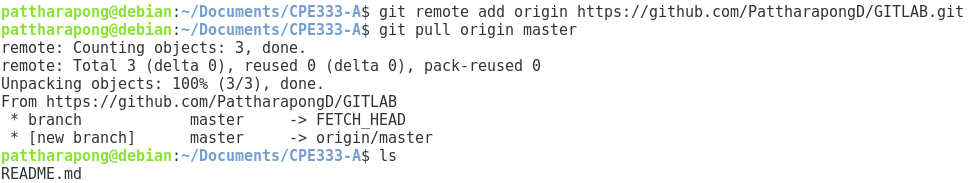




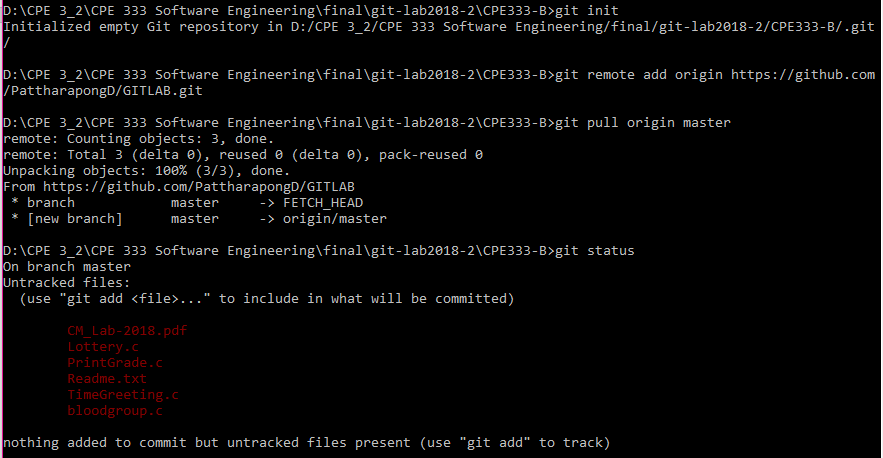


1. In some path create a folder called “CPE333-A” for User A. Make this a git repository and pull data from GitHub. Show status.

https://scontent.fbkk13-1.fna.fbcdn.net/v/t34.0-12/29138341_1824019324335482_667852181_n.png?_nc_cat=0&oh=a97afe9e633cd329ed24601499936e35&oe=5AB76AF1



1. In some path create a folder called “CPE333-B” for User B. Make this a git repository and pull data from GitHub. Show status.



1. User A. Use file manager to put 2 C programs (Program #1 and Program #2), and 1 MS Word document into CPE333-A.
2. User A. Add all files to git. Then commit. Show Status.
3. User A. Push your local repository to the GitHub repository. (You should always pull before you push). Show screen.
4. User B pulls data from GitHub. Show status.
5. User B change 1 function in Program #2. Print local status User B.
6. User B create a new C program Program #3. Do not put into Git yet. Print local status User B.
7. User B adds Program #3 to Git control (git add). Print local status User B.
8. User B commits changes. Print local status User B.
9. User B pushes changes to the GitHub repository. (You should always pull before you push). Print hub status. Print User B status.
10. User A change something in C Program #1. Print local status User A.
11. User A stages its changes and commits at once. Print local status User A.
12. User A pulls repository from GitHub.
13. User A pushes changes to GitHub repository. Print status User A.
14. User B pulls repository from GitHub.
    * All 3 repositories have same copy now.
15. 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.
16. 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.
17. Then User A pulls the changes. Show the Program#1 for both users that they are now the same.