CSCI 3308 Software Development Methods and Tools [Spring 2018]
Instructor: Alan Paradise
TA: Nikhil Sulegaon, Prasanna Srinivasachar, Pratima Sherkane, Rashmi Shetty, Vivian Lai

Version Control

Objectives
- Create a Git repository
- Work with another person and a Git repository
- Understand the basic functionality of Git/version control software

Pre-lab work
If you are not familiar with Git, then run through the Interactive (15 Minute) Git Tutorial (https://try.github.io/levels/1/challenges/1).

Learning Git: Complete the following actions:

Each student works on their own repository until step 14.

1. If you do not already have one, create a GitHub account
2. Create a *public* git repository
3. Clone the repository to your local machine
4. Add and commit a simple README file with your name and the name of this assignment.
5. Add and commit a few other files (anything you want) to the repository
6. Create a tag named v0.0.1
7. Push all changes and tags to your remote repository (note that tags require a special flag to be included in the push)
8. Create and checkout a new branch named testing-new-files
9. Add and commit a few more files to this branch
10. Checkout the master branch
11. Edit and commit at least one of the files in this branch
12. Merge the testing-new-files branch into the master branch
13. Checkout the testing-new-files branch
14. Push all of your changes in both branches to your remote repository (make sure that both branches show up on your remotes host)
15. Find a partner to work with
16. Grant this person/these people write access to your remote repository
17. Clone your partner's remote repository to a new directory outside your original repository
18. Edit the README file in your partner's master branch to list your name as 'Partner: <Your name here>'
19. Commit your changes
20. Push your changes to your partner's remote repository

21. Return to your original repository (but do not pull from your remote branch yet)
22. Checkout your master branch
23. Edit your README file to add a line that says "This might cause a merge conflict"
24. Commit your changes.

25. **Make sure that your partner has completed step 20 (pushed their changes to your README)**
26. Attempt to pull your partner's changes to your repository
27. Resolve the merge conflict if one occurs
28. Commit the merge (assuming a conflict occurred)
29. Copy and paste the entry showing your merge resolution from the log into a text file
30. Push the merged changes back into your remote repo

Credit: To get credit for this lab exercise, show the TA your lab completion log which includes:
- Your name – Github account name
- Your partner's name – Github account name
- The screenshots of the merge conflict and resolution.