# 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:**

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 like) to the repository
6. Created 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 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. Edit and commit at least one of the files in this branch
15. Push all of your changes in both branches to your remote repository (make sure that both branches show up on your remote host)
16. Find a partner or teammate
17. Grant this person write access to your remote repository
18. Clone your partner's remote repository to a new directory outside your original repository
19. Edit the README file in your partner's master branch to list your name as 'Partner: '

20. Commit your changes
21. Push your changes to your partner's remote repository
22. Return to your original repository (but do not pull from your remote branch yet)
23. Checkout your master branch
24. Edit your README file to add a line that says "This might cause a merge conflict"
25. Commit your changes
26. Make sure your partner has completed Step 21 (e.g. pushed their changes to your README)
27. Attempt to pull your partner's changes to your repository
28. Resolve the merge conflict if one occurs
29. Commit the merge (assuming a conflict occurred)
30. Push the merged changes back to 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.