George Mammarella

Lab 2 Questions

1. SVN and Git, both for classes.
2. Not git bash but Ubuntu bash, Windows cmd prompt for PLC, etc. Class stuff.
3. Tells git that you want the added files to be part of your commit. DOES NOT commit.
4. Creates the commit (with included options such as a message) to be sent. Does not send a commit.
5. Pushes (sends) the commit to the master, making changes available to everyone on the repo.
6. Two people. Three repo copies (two teammates + master on GitHub)
7. Two commits by us, one by Sriram. Total three.
8. Atniptw
9. Change README (the change to the README.md file)
10. Two members. Three branches.
11. One on master. One on atniptw (original student). Two on mammargs.
12. Gives you a branch off of the current master which is like your own image of it. You can modify it, pull/push, etc. but it remains in your separate branch until later when you can merge them.
13. Checkout actually switches to a different branch (like the one we previously created with branch)
14. Two members. Three README versions, one on master, one for atniptw, one for mammargs.

Tom and I got a little out of order and he pushed his change to master and I pulled from master (not realizing I had and needed the original clone I made). Because of this, we did not create a conflict.

15. Two members. Atniptw and mammargs were both fast forward. However, if we had made a conflict like we were supposed to, mammargs should have been a manual merge.
16. Three branches, master, atniptw, mammargs
17. Mammargs is because I pulled, no one else made changes, then I merged. Atniptw is not because he made the changes, pushed, then I took it from there. I believe something like this would have occurred even if we had done things in the right order.