Git Homework 2

**Branching**

A key feature to Git is branching. A branch is the ability to break from an existing a branch, make changes, and then merge back into working tree. This is extremely useful for creating new features or fixes because each branch has a name attached to it which can effectively describe the work that was done.

With Github, you can create “protected” branches. A protected branch prevents a user from automatically merging other branches into the protected branch. It is important to prevent “master” from being automatically merged into. Preventing automatic merging is good for the software development lifecycle because people will need to test/review your code before you merge your changes.

For more information on branching, there more resources below:

ARTICLE: <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

VIDEO: <https://youtu.be/JTE2Fn_sCZs>

**Exercise 2.1: Protecting the “master” branch**

1. Login to your Github account and go to the repository you forked from *Git Homework 1.*
2. On the forked repository’s page, click “Settings” which is on the bar directly below repository’s title. (When initially viewing a repository, it defaults to the “Code” Tab)
3. In the “Settings” tab, click “Branches.”
4. Click “Add Rule.”
5. Fill in the “Apply rule to” text box with the branch name that the rule will apply to. In your case it will be “master.”
6. Check off “Require pull request reviews before merging” and “Require status checks to pass before merging.”
7. Click “Create” at the bottom.
8. Take a screenshot of the new rule you have created. Please make sure your Github username is in the screenshot. This will be the submission for this section.

**Exercise 2.2: Creating a new branch**

1. Open your command line or terminal andcdinto the repository.
2. Run agit statusto see where you currently are located in your tree. (Should be “master”)
3. Create a new branch using the command git checkout –b <\*\* branch name \*\*>. This will create a branch directly off of “master” in your local repository. The command to go to other branches that already exists is git checkout <\*\* branch name \*\*>.
4. Add a file, make changes to it in the editor of your choice, stage the file, and now finally commit the file. (If you forget how to do any of this, see Git Homework 1 for information on how to do these tasks.)
5. Since this branch was created in your local repository, Github has no idea that it exists yet. You must tell Github about it by using the command

git push --set-upstream origin <\*\*your branch name\*\*>. This effectively let’s Github watch your branch. This command is necessary when you create a branch on your local repository.

1. Run a final git statusto see if you have any uncommitted/unstaged changes.

**Exercise 2.3**

1. Go back to Github.com and view your forked repository where you just pushed the branch to.
2. Next to the “New Pull Request” button there is a dropdown which displays the current branch you are on. Click that dropdown to see other branches currently in the repository. Your new branch should be listed. Click it to view it.
3. Click the “New Pull Request” button. It should take you to a screen where it shows at the top “base: master” and “compare: <\*\* your branch name \*\*>”. This means that you are trying to merge your branch into master. You can change these if your base and/or compare is not set to the right branches.
4. Enter a thoughtful title for the Pull Request and a short description. In the real world, descriptions can be massive.
5. Click “Create Pull Request” and then go to the “Pull Requests” tab. You should have one Pull Request in there.
6. Click the pull request in the Pull Requests tab and view it. (Take a screenshot of the Pull Request. This is your submission for this section.)
7. Normally, someone will have to review your pull request before it is merged but this is not one of those cases. Below the list of all of your commits, click “Merge Pull Request”. (Do not delete the branch after you click this button.)
8. Your new branch has been successfully merged into “master.”
9. Your master branch in your local repository is now no longer up-to-date with the changes from Github. Open the command line or terminal and go to the “master” branch. (Hint: See 2.2 #3)
10. You must pull your new changes from the repository on Github. To do this, simply run the commandgit pull. Your repository will now be up-to-date with the remote repository.

*SUBMISSION*

1. Submit the link to the forked repository on Canvas which should have your new merged into master. Do not delete this branch after it is merged.
2. Submit the screenshots in a single PDF.