**Ivan Tang**

**CS389**

**Exercise on GitHub and Git**

In this course, we will be using GitHub for assignment submissions.

The goal of this exercise is to get you started with Git and GitHub. Even if you are using Git and GitHub regularly you need to do this exercise and submit it. If you already did it, you still need to create the required repositories and fill out the required forms!

Please follow the instructions completely. Work that does not follow the instructions (including naming conventions) will NOT be accepted and will result in a grade of 0.

**Part 3:**

Answer the following questions.

What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform? (Answer between 5 and 10 lines)

GitHub is a web-based hosting service of Git. It is an online repository, allowing users to upload, download, and make contributions to basically whatever they choose. While used predominantly for computer code, images and text files can still be uploaded. Unlike Git, which is a command line tool, GitHub provides a web-based graphical interface, allowing for easier use. It also provides access control and a version control system, which allows users to download newest version of software, make changes, and upload their updates.

Development for GitHub began on October 19, 2007. It was officially launched in April 2008, by Tom Preston-Werner, Chris Wanstrath, P. J. Hyett and Scott Chacon. GitHub was created to make sharing code easier, whether it be securely or open sourced. GitHub made using Git and sharing code more accessible and easier, influencing more students.

Similar platforms exist, such as BitBucket, SourceForge, GitLab, and BeanStalk. However, GitHub is more widely used. Platforms such as GitHub are used for many reasons. Working on a code together through GitHub is made easy as members have latest access to versions as well as a short summary they can use. Documentation is very important and it is also vital that latest versions are downloaded to be more efficient. GitHub also allows you to show your work off to other people and share software for other people to use.

**Part 4:**

Go through the Git tutorial here: <https://try.github.io>.

Git Tutorial :

Press enter to submit commands

> git init

Initialized empty Git repository in /.git/

Success!

$ git status

# On branch master  
#  
# Initial commit  
#  
nothing to commit (create/copy files and use "git add" to track)

Success!

$ git status

# On branch master  
#  
# Initial commit  
#  
# Untracked files:  
# (use "git add <file>..." to include in what will be committed)  
#  
# octocat.txt  
nothing added to commit but untracked files present (use "git add" to track)

Success!

$ git add octocat.txt

Nice job, you've added octocat.txt to the Staging Area

$ git status

# On branch master  
#  
# Initial commit  
#  
# Changes to be committed:  
# (use "git rm --cached <file>..." to unstage)  
#  
# new file: octocat.txt  
#

Success!

$Ggit commit -m "Add cute octocat story"

[master (root-commit) 20b5ccd] Add cute octocat story  
1 file changed, 1 insertion(+)  
create mode 100644 octocat.txt

Success!

$ git add '\*.txt'

$ git commit -m "Add all the octocat txt files"

[master 3852b4d] Add all the octocat txt files  
4 files changed, 4 insertions(+)  
create mode 100644 blue\_octocat.txt  
create mode 100644 octofamily/baby\_octocat.txt  
create mode 100644 octofamily/momma\_octocat.txt  
create mode 100644 red\_octocat.txt

Success!

$ git log

commit 3852b4db1634463d0bb4d267edb7b3f9cd02ace1  
Author: Try Git <try\_git@github.com>  
Date: Sat Oct 10 08:30:00 2020 -0500  
  
Add all the octocat txt files  
  
commit b652edfd888cd3d5e7fcb857d0dabc5a0fcb5e28  
Author: Try Git <try\_git@github.com>  
Date: Sat Oct 10 08:30:00 2020 -0500  
  
Added cute octocat story

Success!

$ git remote add origin https://github.com/try-git/try\_git.git

Success!

$ git push -u origin master

Branch master set up to track remote branch master from origin.

Success!

$ git pull origin master

Updating 3852b4d..3e70b0f  
Fast-forward  
yellow\_octocat.txt | 1 +  
1 file changed, 1 insertion(+)  
create mode 100644 yellow\_octocat.txt

Success!

$ git diff HEAD

diff --git a/octocat.txt b/octocat.txt  
index 7d8d808..e725ef6 100644  
--- a/octocat.txt  
+++ b/octocat.txt  
@@ -1 +1 @@  
-A Tale of Two Octocats  
+[mA Tale of Two Octocats and an Octodog

Success!

$ git add octofamily/octodog.txt

Success!

$ get diff --staged

diff --git a/octofamily/octodog.txt b/octofamily/octodog.txt  
new file mode 100644  
index 0000000..cfbc74a  
--- /dev/null  
+++ b/octofamily/octodog.txt  
@@ -0,0 +1 @@  
+[mwoof

Success!

$ git reset octofamily/octodog.txt

Success!

$ git checkout -- octocat.txt

Success!

$ git branch clean\_up

Success!

$ git checkout clean\_up

Switched to branch 'clean\_up'

Success!

$ git rm '\*.txt'

rm 'blue\_octocat.txt'  
rm 'octocat.txt'  
rm 'octofamily/baby\_octocat.txt'  
rm 'octofamily/momma\_octocat.txt'  
rm 'red\_octocat.txt'

Success!

$ git commit -m "Remove all the cats"

[clean\_up 63540fe] Remove all the cats  
5 files changed, 5 deletions(-)  
delete mode 100644 blue\_octocat.txt  
delete mode 100644 octocat.txt  
delete mode 100644 octofamily/baby\_octocat.txt  
delete mode 100644 octofamily/momma\_octocat.txt  
delete mode 100644 red\_octocat.txt

Success!

$ git checkout master

Switched to branch 'master'

Success!

$ git merge clean\_up

Updating 3852b4d..ec6888b  
Fast-forward  
blue\_octocat.txt | 1 -  
octocat.txt | 1 -  
octofamily/baby\_octocat.txt | 1 -  
octofamily/momma\_octocat.txt | 1 -  
red\_octocat.txt | 1 -  
5 files changed, 5 deletions(-)  
delete mode 100644 blue\_octocat.txt  
delete mode 100644 octocat.txt  
delete mode 100644 octofamily/baby\_octocat.txt  
delete mode 100644 octofamily/momma\_octocat.txt  
delete mode 100644 red\_octocat.txt

Success!

$ git branch -d clean\_up

Deleted branch clean\_up (was ec6888b).

Success!

$ git push

To https://github.com/try-git/try\_git.git  
3e70b0f..33a28d4 master -> master

Success!

**Part 5:**

Define the following terms in the context of Git (2 lines maximum):

* Repository – where files are kept
* Commit – stores changes to local repository
* Push – sends the changes to a remote repository
* Branch – copy of code made within the repository that developers make to make different changes
* Fork – makes a clone of a repo and puts it in your account so you can make further changes to someone’s project
* Merge – take the changes made in one branch and apply it to another
* Clone – create a local copy of a git repo
* Pull – when you add the changes from a remote repo to your local one
* Pull request – proposed changes to a repo that is suggested by a user, and the repo developer’s choice whether or not to commit the changes

**Part 6:**

Push the Word file in **YOUR** GitHub account in a repository called ***CSXXX20XX***. Please respect the naming conventions! You will use this repository this semester. Your repository will be accessible at: [https://github.com/yourpseudo/CSXXX20XX](https://github.com/yourpseudo/CSXXX2016).

**Part 7:**

Forked to my GitHub

Edited README.

Committed to master branch.