**Exercise on GitHub and Git**

**Mobile Innovations for Global Challenges**

**Tanya Sahin**

**Part 1:**

My GitHub account name is t-a-n-y-a. I submitted it in the Google Docs form.

**Part 2:**

I successfully installed Git bash.

**Part 3:**

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 an internet-based Git repository hosting service that provides a graphical interface (in contrast to Git, which is purely command-line based). Git itself is a Distributed Version Control System (DVCS), created in 2005 by the Linux development community, in particular Linus Torvalds. Git was developed in need of a DVCS, after the previously used proprietary DVCS, BitKeeper wanted to start charging Linux. Similar platforms are Mercurial, Bazaar, or Darcs. Such a platform is very useful to save different versions of projects. It enables collaboration on the same project (code, graphics, writings) and allows easy going back to an older version in case errors are made. Further, each collaborating member usually has all versions of the whole project, thus the platform also reduces the risk of data loss.

**Part 4:**

Go through the Git tutorial here: https://try.github.io. While doing the tutorial, save your work the LastnameFirstnameGitTutorial-mm-dd-yyyy.docx file.

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!

$ git 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'

Success!

$ git status

# On branch master

# Changes to be committed:

# (use "git reset HEAD <file>..." to unstage)

#

# new file: blue\_octocat.txt

# new file: octofamily/baby\_octocat.txt

# new file: octofamily/momma\_octocat.txt

# new file: red\_octocat.txt

#

Did not use git commit

$ 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!

$ git 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 branch

clean\_up

\* master

Use 'git checkout' to switch to the 'clean\_up' branch

$ 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..e1e5078 master -> master

Success!

>

**Part 5:**

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

• Repository

This is the database were all history and configuration is stored. It may contain several branches.

• Commit

A commit is a ‘snapshot’ of your project at a certain time. As a verb, it means the action of storing a new snapshot in the Git history.

• Push

Pushing refers to sending your committed changes to a remote repository

• Branch

A different line of development within the project.

• Fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

• Merge

To merge means to bring the contents of another branch (possibly from an external repository) into the current branch.

• Clone

A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy.

• Pull

Pulling a branch means to fetch it and merge it.

• Pull request

Pull requests let you tell others about changes you've pushed to a repository on GitHub. Once a pull request is sent, interested parties can review the set of changes, discuss potential modifications, and even push follow-up commits if necessary.

Sources:

<https://www.kernel.org/pub/software/scm/git/docs/gitglossary.html>

<https://help.github.com/articles/github-glossary/>

<http://stackoverflow.com/questions/7076164/terminology-used-by-git>

**Part 6:**

I successfully pushed this document to my GitHub CS6432015 repository.

**Part 7:**

I forked the repository at https://github.com/paceuniversity/courses.

Then, I entered the following commands:

**git clone** [**https://github.com/t-a-n-y-a/courses.git**](https://github.com/t-a-n-y-a/courses.git)

Then, I added my name followed by the date and time to the README.md file.

Next, I added, committed, and pushed the changes:

**git add README.md**

**git commit –u "Added my name Tanya Sahin"**

**git push -u origin master**

After successfully pushing, I went on the GitHub Website into my courses repository and clicked on “pull requests”, then clicked “new pull request”, and then submitted the request.