Kenneth Mann

CS 611 – Github Assignment

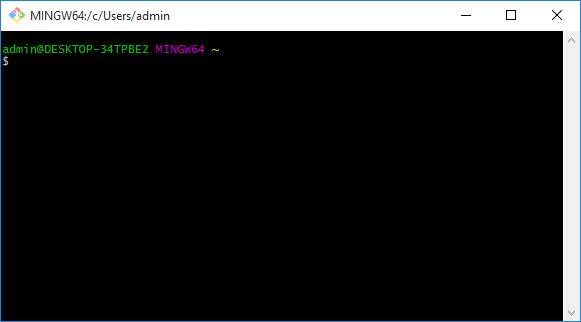
**Part 1:**

GitHub Username: KenMannPace

GitHub Email: [km30566n@pace.edu](mailto:km30566n@pace.edu)

**Part 2:**

Install Git bash to local laptop (Done)



**Part 3 - Questions to answer:**

**What is GitHub?**

Github is software that can be used for version control. Version control refers to the process whereby software versions are kept in an orderly fashion and can be referred back to or ‘rolled back’ if software of a previous version is needed. Github has the added feature that it runs on a remote web platform and so does not need to run as a local network application (Ref: #1).

**When was it created?**

Github was founded in February 2008. (Ref: #2).

**Why?**

From reading one of the founders, it seems that Github was written in order to make a centralized place where people could use “Git”, version control software, yet then to be able to, quoting the founder blog “securely share private code”. (Ref: #3).

**By who?**

PJ Hyett, Chris Wanstrath, Tom Preston-Werner (Ref: #4).

**What similar platforms exist?**

GitHub, Google Code, Bitbucket, CodePlex and SourceForge are examples of other version control systems (found by looking up “cloud based version control systems on Google (Ref: #5).

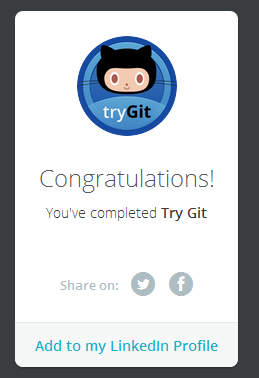
**Why would you use such a platform?**

In general, a version control system is something like if you had a folder on your thumb drive where you saved numbered and dated versions of a piece of software that you were periodically updating. If you ever needed to go back to an older version of the software, then you could use the backup disk copy to do so. (Ref: #6).

**Part 4 – Git tutorial:**

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

I registered with CodeSchool and received this merit badge for completing the tutorial. I did not see a way to save the material.



**Part 5 - Define the following terms in the context of Git:**

**Repository:**

A repository is a place in Github that allows you to organize the particular software you are creating. For example, you can store pictures, code files, etc., basically anything that can be saved as a file and it will be organized under the related repository name. (Ref: #1).

**Commit:**

On Github a commit is simply a name for saving changes. However, recall that a version control system tracks changes, so there can be multiple versions of the file (unlike just saving changes to a Word document which would over-write the previous state of the document). (Ref: #1).

**Push:**

Pushing is the act of sending along changes to a document to a repository which is not on the local machine (I would imagine usually GitHub itself?). (Ref: #7).

**Branch:**

A branch is a copy of the “master” version of the repository and is stored within that master repository. Branches can be used to try out code, etc. before it is committed to the master-branch. (Ref: #7).

**Fork:**

A fork is a copy of another user’s account’s repository that is then connected to your personal Github account. There connection remains back to the user’s account from whence the connection arose. One can pull updated data from the original repository and vice versa. (Ref: #7).

**Merge:**

A merge operation will apply the changes from one branch into a destination branch. (Ref: #7).

**Clone:**

A clone is a copy of a remote repository (remote meaning the web cloud) that has been copied to the local machine. This copy can be edited as you like, locally and then, if desired, it can be sent back to the original repository (in a synch-type operation). (Ref: #7).

**Pull:**

A pull operation refers to the actual, in progress, act of downloading change requests that have been made and merging the changes into…(repositories, branches, etc.). (Ref: #7).

**Pull request:**

A pull-request is the pre-approval aspect of a pull. When one wishes to conduct a pull, one will send a pull-request to those in charge of the repository for approval. (Ref: #7).

**Part 6:**

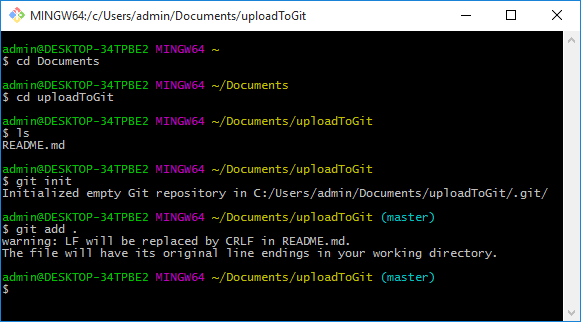
Push the Word file in YOUR GitHub account in a repository called CSXXX2016. Please respect the naming conventions! You will use this repository this semester. Your repository will be accessible at: <https://github.com/yourpseudo/CSXXX2016>.

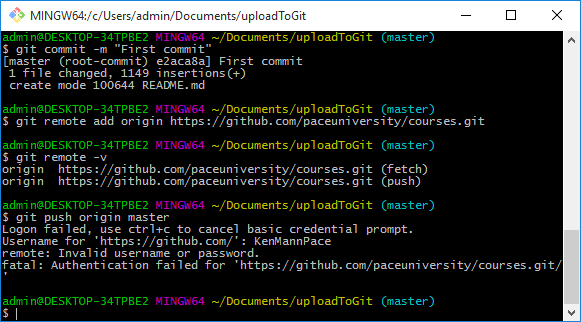
Part 7:

Retreival and push of class Readme file:

\*\*I **retrieved** and **edited** the specified Readme file from github.com/paceuniversity/courses

Yet I have not ***yet*** succeeded in pushing it back… (please see below)





(Ref: #8).

**Part 8:**

Add an issue with title “GitHub training” in your repository called CSXXX2016. Issues will be used for tasks and bug reports. (Ref: #9)

**(DONE)**

**Part 9:**

Edit the main page of the wiki in your repository called CSXXX2016. Add the title “CS XXX 2016” to the page. The wiki will be used for documenting your work.

**(DONE)**

**References**

1) https://guides.github.com/activities/hello-world/

2) https://en.wikipedia.org/wiki/GitHub

3) http://tom.preston-werner.com/2011/03/29/ten-lessons-from-githubs-first-year.html

4) https://www.crunchbase.com/organization/github#/entity

5) https://blog.idrsolutions.com/2014/03/top-5-free-hosted-version-control-sites-compared/

6) http://mikemcquaid.com/2014/01/18/why-use-version-control/

7) https://help.github.com/articles/github-glossary/

8) https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/

9) https://help.github.com/articles/creating-an-issue/