**CS 639 Summer I 2019**

**Answers for Exercise on GitHub and Git**

**Part 3**

What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform?

**Answer:**

* **Git** is a distributed version-control system for tracking changes in source code during software development, whereas **Github** is code sharing and version control **service** for programmers to **remotely** **collaborate** with multiple people on the project. It offers all the distributed version control and source code management (SCM) functionality of Git. It also provides features like bug tracking, feature requests, task management.
* Github was created to facilitate reliable hosting of Gits online and easy collaboration. Github aims to offer free and inexpensive hosting of code.
* Github was founded in 2008 by Tom Preston-Werner, Chris Wanstrath, P. J. Hyett and Scott Chacon.
* There exist many code hosting platforms like CVS (1986), Subversion (2000), Mercurial, BitKeeper.
* Reason for using such platform -
  + - * For better and easy software configuration management.
      * To protect code from physical damages or human errors.
      * To see the entire timeline of the code changes, decisions, and progression of any project in one place.
      * To synchronize work of multiple developers without wasting hours merging.
      * To find when a bug was introduced easily.

**Part 5 : Defining terms in context for Git**

**Answer:**

* Repository

A repository encompasses the entire collection of files and folders associated with a project, along with each file’s revision history.

* Commit

Git commit saves the snapshot to project history and completes the change tracking process. Anything that is staged by ‘git add’ will become part of snapshot once ‘git commit’ is used.

* Push

This is used to push the local changes/commits to remote repository so that changes are available for all other collaborating members

* Branch

Branching of a project helps to try out new ideas/features without affecting the master branch. This gives us freedom to experiment and commit changes, safe in the knowledge that our branch won't be merged until it's ready.

* Fork

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

A connection always exists between the fork and the original repository itself so that one can contribute back to original project using pull requests.

* Merge

Merge is used when we work on branch and is now ready to be merged into master branch. To merge changes to master, first we would have to check out the master and then run merge command to merge the changes to master

* Clone

Cloning also lets us make local copy of the repository but it’s ideal for instances when we need a way to quickly get our own copy of a repository where we may not be contributing to the original project.

* Pull

This is used to pull changes from remote repository into personal local repository

* Pull request

Pull requests are made to another repository to ask their maintainers to pull our changes into theirs.

**Part 7 : Strategy used to update file at** [**https://github.com/paceuniversity/courses**](https://github.com/paceuniversity/courses)

**Answer:**

1. Created a fork by clicking on fork button on the master repository <https://github.com/paceuniversity/courses>
2. Then cloned the fork using Github desktop application
3. Made changes to the README.md file locally
4. Staged and pushed the changes using Git Bash into forked version of the repository

* cd <directory where the cloned copy resides>
* git status
* git add README.md
* git commit -m “added sukhada sheth”
* git push

1. Created a pull request using Pull Request button for professor to pull the changes which I made from my forked version into the master copy of professor.