**PART 3:**

Answer the following questions.

What is GitHub? GitHub is a web based Git repository hosting service. It offers revision control and source code management functionality of Git but it also adds its own features. It is a web based graphical interface.

When was it created? The site launched in April 2008.

Why? The site launched to make Git more accessible to person who are not familiar with Linux Kernel.

By who? Tom Preston-Werner, Chris Wanstrath, and PJ Hyett.

What similar platforms exist? Assembla, BitBucket, CloudFore, Gitorious, Unfuddle, and others will host remote repositories for you.

Why would you use such a platform? It allows registrered and non registered users to browse public repositories on the site. You have multiple persons working on a single projects from different sites. Each member branches out from the main master branch. They want to have one area where they can keep track of what work they are doing. Git allows you to create copies of the main project and commit them to the main repository. That way, everyone will be on the same track. GitHub uses Git to complete the same task. (Answer between 5 and 10 lines)

**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.

1. Initialization: git init
2. Checking status: git status
3. Adding and committing: git status
4. Adding changes: git add octocat.txt
5. Checking changes: git status
6. Commiting: git commit –m
7. Add all changes: git add ‘\*.txt’
8. Commiting all changes: git commit –m ‘Add all the octocat txt files’
9. History: git log
10. Remote repositories: git remote add origin https…
11. Pushing remotely: git push –u origin master. –u is to remember the parameters
12. Pulling remotely: git pull origin master
13. Differences; git diff HEAD
14. Staged differences: git add octofamily/octodog.txt
15. Staged differences: git diff –staged
16. Resetting the stage: git reset to unstage files
17. Undo: git checkout – filename
18. Branching out: git branch clean\_up
19. Switching branches: git checkout
20. Removing all the things: git rm ‘\*txt’
21. Commiting branch changes: git commit –m
22. Switching back to master: git checkout master
23. Preparing to merge: git merge clean\_up
24. Keeping things clean: git branch –d clean\_up. –d is for delete.
25. Final push: git push

**PART 5:**

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

* Repository: a data structure that stores information. Contains a set of commit objects and references to objects.
* Commit: used to record your snapshot into your history. Kind of like save.
* Push: After we crate repository we use push to send our remote repository to the main one. You push changes but you have to provide the actual branch name.
* Branch: list all of the branches in your repo. You can also create a branch by adding an argument.
* Fork: copy of a repository.
* Merge: used to merge branches to current branch. Unify the changes with the primary copy.
* Clone: used to clone repository located in a certain location onto local machine.(All above from Atlassian Git cheat-sheet).
* Pull: fetch specified remotes copy of current branch and merge it into the local copy. .(All above from Atlassian Git cheat-sheet).
* Pull request : this lets you tell others about changes you’ve pushed to a GitHub repository. Interested parties can review changes. (<https://yangsu.github.io/pull-request-tutorial/>).

PART 7:

I had to fork the file, edit it and propose the change I made on it to you.