**Q1) What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform?**

**Ans:** GitHub is a web-based version-control and collaboration platform for software developers. GitHub , which is delivered through a software-as-a-service(SaaS) model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster.

Git is used to store the source code for a project and track the complete history of all changes to that code.

Some similar platforms are:

Source Code System(SCCS)

Revision Control System(RCS)

CVS (Concurrent Version System)

Apache Subversion(SVN)

BitKeeper SCM

Such platforms are used because they facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration .

**Q2) TUTORIAL REVIEW**

**Ans:**

1. **INITIALIZATION:**Once you are in your current directory (here say:”octobox”) you can initialize your git repository (is a hidden directory where git operates)here.

**git init**

1. **STATUS:**To seethe current status of your project use

**git status**

1. **ADDING AND COMMITING:** You can add by adding a text file(say octocat.txt) in your repository which can be seen using status of your git.
2. **ADDING CHANGES:** Git only starts tracking changes made to your text file(here octocat.txt) once you have added it to your staging area.

**git add octocat.txt**

1. **CHECKING CHANGES:** git status
2. **COMMITTING:** Once the files are in Staging area they are ready to be added to your repository you run commit command with a message of the changes that you have made.

**git commit –m “Changes in octocat are these”**

1. **ADDING ALL CHANGES:** To add all the files to the staging area use

**git add ‘\*.txt’**

1. **COMMITTING ALL CHANGES:** After adding files to staging area you can commit them using

**git commit –m ‘Add all the content of the files’**

1. **HISTORY:** To see all the changes commited so far we can use git log

**git log**

1. **REMOTE REPOSITORY:** To push your local repo to the GitHub server we need to add a remote repository. This command takes a remote name and a repository URL

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

1. **PUSHING REMOTELY:**  PUSH command tells Git where to put our commits when we are ready.

Eg): The name of our remote is **origin** and the default local branch name is **master** .The **–u**  tells Git to remember the parameters .

**git push –u origin master**

1. **PULLING REMOTELY:**  To check for changes on our GitHub repository and pull down any new changes.

**git pull origin master**

1. **DIFFERENCES:** To get the difference of our most recent commit and previous commits, using HEAD pointer.

**git diff HEAD**

1. **STAGED DIFFERENCE:** Diff is also used for looking at changes within files that have already been staged(Files ready to be commited).

**git diff --staged**

1. **RESETTING THE STAGE:** To unstage files use git reset command.

**git reset octofamily/octodog.txt**

1. **UNDO:** Files can be changed back to how they were at last commit .

**git checkout –-octocat.txt**

1. **BRANCHING OUT:**  If you want to create a copy of your wirk or you can create a seprate branches and commit to it as and when needed, say you create branch named clean\_up.

**git branch clean\_up**

1. **SWITCHING BRANCHES:**  If you have more than one branch say Master branch and clean\_up branch and you want to commit to both as per requirements then you can keep switching between the to branches.

**git checkout clean\_up**

1. **REMOVING CONTENTS:** If you want to remove a content use rm it not only removes the content from actual disk but will also stage the removal.

**git rm ‘\*.txt’**

1. **COMMITING BRANCH CHANGES:** To commit any changes in the branches use

**git commit –m “Remove changes”**

1. **SWITCHING BACK TO MASTER:** If you want to switch from clean\_up branch to master branch so you can copy changes from clean\_up to master

**git checkout master**

1. **MERGE:** If you want to add your changes from your clea\_up branch into your master branch you can merge them.

**git merge clen\_up**

1. **DELETING BRANCH:** To delete a branch completely.

**git branch –d clean\_up**

1. **FINAL PUSH:**Now everything you have done has to be pushed onto your remote repository.

**git push**

**Q3) Explain the following terms:**

**Ans:**

1. **REPOSITORY:** The purpose of Git is to manage files as they change over time. Git stores this information in a data structure called a repository. A git repository contains a set of commit objects and a set of references to commit objects called heads.
2. **COMMIT:** basically git commit puts your changes into your local repository.
3. **PUSH:** git push is used to add commits you have done on the local repositoryto a remote one.
4. **BRANCH:** A branch represents an independent line development.It reuests a new working directory,staging area, and project history.
5. **PUSH:** Updates remote refs using local refs, while sending objects necessary to complete the given refs.
6. **FORK:** A fork is a copy of a repository . Forking a repository allows you to freely experiment with changes without affecting the original project.
7. **MERGE:** This command is used by git pull to incorporate changes from another repository and can be used by hand to merge changes from one branch into another.
8. **CLONE:** When you create a repository on Git it exists as a remote repository. You can clone your repository to create a local copy on your computer and sync between the two locations.
9. **PULL:** Incorporate changes from a remote repository into the current branch.
10. **PULL REQUEST:** It will tell others about changes you have pushed to a repository on Git. Oncd a pull request is opened, you can discuss and review the potential changes will collaborators.

**STEPS FOR EDITING README FILE ARE AS FOLLOWS:**

STEP 1: Create a Clone of your Repository using the Fork command from Dashboard.

STEP 2: Copy your document file into your current working Directory.

STEP 3: Add and Commit your file into the Repository.

STEP 4: Push origin master into the git repository.