**Git - 3 Days (3 x 4 = 12 Hours)**

**Day 1**

* The need for a Version Control System
* Centralized vs Distributed VCS
* What & Why Git?
* Installing Git
* Using *git config* command to set username & email address

Link: <https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup>

* What is a Git Repository?
  + Local vs Remote repo
  + A very brief introduction to Git hub
    - ***Demo of how to create an empty public remote Git Hib repo***

Link: <https://git-scm.com/book/en/v2/GitHub-Account-Setup-and-Configuration>

* Cloning a remote repo
  + git init
  + git clone vs git remote add
* Basic Git commands
  + *git status*
  + Understanding File stages
  + *git add*, *git commit* & *git push* commands

**ASSIGNMENTS**

1. Create a remote git hub repository using Git Hub*. (you can use any email id for creating)*
2. Clone the repository on your local machine.
3. Using *git config*, set the username & email to your name and your CT email id respectively.
4. Create two empty text files named **f1.txt** and **f2.txt** inside the cloned repository.
5. Commit and push the changes to the remote repository.
6. Check the status of the local repo.
7. Go to the remote repository & check what files are present.
8. Add some text in **f2.txt** and save the changes.
9. Check the status of the local repo.
10. Go to the remote repository & check the contents of **f2.txt.** What do you see and why?
11. Add the file **f2.txt** into staging.
12. Check the status of the local repo.
13. Go to the remote repository & check the contents of **f2.txt.** What do you see and why?
14. Use appropriate Git commands to push the changes to the remote repository.

**Day 2**

* Checking commit history
  + *git log* command

Link: <https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History>

* Undoing
  + Using the *-amend* option with the *git commit* command
  + Using the *git reset* command

Link: <https://git-scm.com/book/en/v2/Git-Basics-Undoing-Things>

* Introducing Branching
  + What & Why branches?
  + What is the *master* branch?
  + Creating a new branch
    - *git branch* command
  + The *HEAD* pointer
  + Switching between branches
    - *git checkout* command

Link: <https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

**ASSIGNMENTS**

1. Continuing from the previous assignment, make sure that there are no files in the staging area & that the remote repository contains files **f1.txt** and **f2.txt** with some content.
2. Create 3 new files named **f3.txt, f4.txt** and **f5.txt** respectively.
3. Commit the changes. *(do not push)*
4. Add some text again in **f1.txt** and save the changes.
5. Stage the changes made to **f1.txt.**
6. Use the *commit –amend* command to undo the last commit.
7. Push the changes to the remote repo finally. Go to the remote repo and check the results.
8. Create a new branch named **branch1.**
9. Switch to this branch.
10. Add some lines in the file **f1.txt** and save the changes.
11. Stage and commit. *(do not push)*.
12. Check the contents of the file.
13. Switch back to the master branch.
14. Add some lines in the file **f1.txt** and save the changes.
15. Stage and commit. *(do not push)*
16. Check the contents of the file.
17. From the master branch, push the changes to the remote repository. Go to the remote repo and check the contents of the **f1.txt** file.

**Day 3**

* Significance of *git pull* command
* Merging Changes (https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging)
  + Basic merging (*git merge* command)
  + Basic Merge Conflicts and how to resolve them
* Git rebasing concepts (<https://git-scm.com/book/en/v2/Git-Branching-Rebasing>)
  + Basic rebasing
* Pull request in Github *(this can be given as a self-study)*

Link:

<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>

**ASSIGNMENTS**

1. Form teams of 10 participants each.
2. A member from each team will create a remote repo on GitHub with his/her email id.
3. All other members of the team will clone that remote repo on their laptops.
4. The master branch will contain a file named **main.txt** with some text in it.
5. Each member will create a new branch and add some text in the **main.txt** file.
6. Each member will try to merge changes made by them into master branch. Observe if merge conflicts occur and try to resolve it.