1) GIT Installation:

\* Git Bash

\* Git GUI

\* Download GIT - Folder -D:\Hema\_contents\java\_jee\GIT\Git and add it to the path

D:\folder1>set path=D:\Hema\_contents\java\_jee\GIT\Git\bin;%path%

2) Creating a GIT repository

\* **git init -** full fledged GIT repository with all VC and branching capabilities

\* **git init --bare -** Bare repository in which u cannot create or commit files but others can push their files

3) Status

\* **git status -** The status of any repository can be viewed using this command

It tell u if there are untracked files or uncommited files or if everything is commit

4) GIT will execute a commit not in the folder but in staging area - a temporary area where u will keep files

\* **git add f1.txt** - Only f1.txt is moved to the staging area

\* **git add .** - Move all untracked or uncommited files in the folder to the staging area

5) To commit the file(s) staging area

\* **git commit -m "This is first file"**

**\* git commit -m "V1.0"**

6) To view the history of all commits

**\* git log**

7) We can create a repository by cloning another repository

**\* git clone D:\folder1**

**\* git clone** [**git@github.com:HemaGRepo/JuneD1.git**](mailto:git@github.com:HemaGRepo/JuneD1.git)

8) You can restore the contents of a GIT repository as per any commit – Remember the commit Id:

**\* git checkout <commit Id 7 charcs>**

**\* git checkout 341f584**

9) You can create tags for every commit

hemag@CHNL19924 /d/folder1 ((341f584...))

**$ git tag "T3" 63369c2**

**hemag@CHNL19924 /d/folder1 ((341f584...))**

**$ git checkout T3**

Previous HEAD position was 341f584... Some Java Code Committed

10) You can check how many branches are there in a GIT repository

**$ git branch**

* A Branch is a linear line of development
* You can create a child branch from an existing branch
* Branches can be given names
* When a git repo is created automatically a branch called “master” is created

11) To create a new branch

* **$ git branch Test**

12) To switch over to a branch

* **$ git checkout Test**

13) Merging of branches

* **git checkout Test**
* **You will be in Test branch**
* **Git merge Design**

a) No conflict – Both branches do not have the same file with diff content

b) Conflict - Both branches have the same file with diff content – Manual resolution and commit

14) To view the changes in the last commit use

* **git show**

15) To copy the contents from a branch of one repo to the same branch of another repo

**D:\folder3> Git push D:\folder4 Test**

Here Test branch in folder3 repo is copied to Test branch in folder4 repo

To push contents to a remote repository

**Git remote add origin** [**git@github.com:HemaGRepo/Examples.git**](mailto:git@github.com:HemaGRepo/Examples.git)

**Git push origin Test**

16) A branch which was pushed can fetch the remaining contents and merge it with the existing content using Pull

**D:\folder4> Git pull D:\folder3 Test**