Git topics
========
1. Introduction
2. What is VersionControl System(VCS) and types of VCS
    a. Local Version Control System
    b. Centralized Version Control System
    c. Disturbuted Version Control System(github,bitbucket,...)
3. Git Software installation
4. Git project Architecture
5. Git commands or Operations

```
            1. git help                  8. git commit                15. git fetch
22. git show
            2. git init                    9. git log                    16. git
stash            23. git revert
            3. git config           10. git clone                17. git merge

            4. git add              11. git push                 18. git rebase
            5. git status           12. git pull                       19. git
diff
            6. git rm               13. git branch               20. git revert
            7. git restore          14. git checkout        21. git mv
```

6. Git commands execution
    a. using commandline(gitbash,gitcmd,gitgui)
    b. using IDE's like Eclipse/STS(SpringToolSuite)/Intelij

7. Github account creation
    a. public repository creation
    b. private repository creation

8. Git folder structure

9. Git Branching Strategy
    1. developer branch
    2. master branch
    3. release branch
    4. hofix branch
    5. fork branch

10. Project Review Process(PR process), Code Reviews,Code Merge

11. Realtime problems with git and how to fix them

12. FAQ's

Introduction
==========
=> Git is a popular Version Control system (VCS)
=> It was created by Linus Torvalds in 2005 and it is maintained by Junio Hamano

Git is used for
    a. Tracking code changes
    b. Tracking who made the changes like history of files
    c. Coding Collobarations

What is VersionControl System(VCS) and types of VCS?

It is a system that records changes made to the file or set of files over the time,so that we can recall the
specific version later.
ie, for every source code change in a file a new version will be created
eg: JDK1.0V, JDK1.1V, JDK1.2V,.........
Spring1.X,Spring2.X,Spring3.X,......

Types of Version Control Software(VCS)
There are 3 types of VCS
a. Local Version Control System
b. Centralized Version Control System
c. Distributed Version Control System

Local Version Control System
=> It is used to maintain the file version and retreive the files based on the specific version
refer : LocalVersionControlSystem.png

To overcome the drawback of LocalVersionControl System we have "Centeralized Version Control System".

Centeralized Version Computer

=========================

=> Developers can collobare the code in one repository and do the change.
eg of Centeraized Version softwares: SVN,Subversion, Peforce,......

=> Centeralized Version server will have single server that contains all the version files
=> For many years this has been the standard version control system
=> More no of devleopers would connect to CVS to checkout the files

Note:
Checkout -> taking the code from repository to local machine.
push -> sending the code from local machine to repository(CVS)

Advantage
1. EveryOne know to certain degree what everyone else on the project is doing.
2. Administrator will have full control over whoch can do what and it is easier to manage.

Drawback

----

1. Single point of Failure(SPF) would represent the Centralized system.
2. If the server goes down due to network traffic, during that hour nobody can collobarate at atll or save changes
to the server.
3. If the hard disk of the centralized system gets corrupted and proper backup haven't been taken then there is
every possiblity of loss of data.

Note: In LVCS ad in CVCS getting up the complete history of changes is not possible.
It is possible to get only the latest version,but not the entire history.
eg: SVN
push will not happen w.r.t version rather push will happen only with the latest change.

```
Version history
file -> 1.0V
file-> 1.1V
file-> 1.2V
file-> 1.3V

Distrubuted Version Control System
=============================
      Eg: Git, Mercurial, Darcs, Baazar, etc,....
=> Developers will not only get the latest version but also the compile history of
the files
=> Push will not only happen with latest snapshot of the files rather they will
push the old files also.
=>If the main sever goes off, still there is a local repository which would have
maintained the copy of the repository
    where the entire code is available(history of versions).
=> If the remote repository is down, then devleoper can do changes in the local
repository and when the main
     repository is up the code can be pushed to remote repository from local
repository.


 Git installation
 -------------------
There are 2 types of Git software
            1. Git Server
            2. Git Client

GitServer
=======
      -> It is a repository
      ->  It is the largest host of source code in the world.
      -> It is used to store/manage the source code of the project
      -> Some of the Git server tools are : Github,BitBucket,GitLab,.......

                  refer :gitserverarchitecture.png

Where should we provide url, username and password?
     To type these details we need git client.

GitClient
=======
Installation of git software
   1. Download a git software from the following link
                  https://git-scm.com/download/win

   It is a tool which is used to connect to our gitserver.
   if we install git client(git s/w) we get the following tools for free
            a. git bash => linux commands  are required
            b. git gui    => Graphical user interface where all the actions will be
done through clicks
            c. git cmd  => command line tools where developer should provide
url,username and password

Note:
   gitclient is a .exe file which can installed with just few clicks.
   git -> client tool where the client should provide url, username and password
   github-> server software where repositores/projects will be maintained
```

```
Git Architecture
---------------------
            refer :gitarchitecure.png

There are 3 regions
        a. workplace => It is a place where devlopers maintain there source code
        b. stage area => Once the code is ready, then it will be added to stage
area(indication to git software)
       c.  local repository=> Once the code is in stage area, we commit it to the
local repository with some standard
                                       message, From local repository we "push" to
main repository by providing
                                         url,username and password.
```