Version Control Softwares

=> Project contains multiple developers

=> Developers will be working from different locations

Problem-1 : How to integrate all the developers code at once place

Problem-2 : How monitor/track code changes

(who , when , why , what)

=> To resolve above problems we will use Version Control Tools...

=> We have several version control softwares

- SVN (outdated)
- Git Hub
- Bit Bucket

Git Hub

=> Git Hub is a cloud platform

=> Using git hub we can maintain source code repositories

=> Source code repos are used to store project source code at once place

Note: For every project one github repository will be created.

-> Git Repository will provide monitored access.

Environment Setup

1) Create account in github

URL : www.github.com

2) Download and install git client software

URl : https://git-scm.com/download/win

3) Configure your name and email in git bash.

```
$ git config --global user.name "Ashok"

$ git config --global user.email "ashokitschool@gmail.com"
```

Git Architecture

1) Working Tree

2) Staging Area

3) Local Repository

4) Central Repository

=====================
Git Repo practicals
=====================

1) Create git repo (public)

Ex: https://github.com/ashokitschool/inst_mgmt.git

2) Copy script from git repo and execute it in git bash

Note: Git bash is a client to perform git operations with github repo

====================
Git Bash commands
====================

git init : To initialize working tree

git status : To check staging area status

git add : To add files to staging area

git commit : Send files from staging area to local repo

git push : Send files from local repo to central repo.

git restore : To discard working tree changes & to unstage the file

git log : To get repo commit history

git rm : To remove file (rm + commit + push)

git clone : To download central repo to local machine

git pull : To get latest changes from central to local

==================================
What is .gitignore in git repo ?
==================================

=> This file is used to specify which files & folders we don't want to commit to git repo

Ex: .project , .settings, target ....

========================
What is git conflict ?
========================

=> When we are merging central repo changes with local repo then we may get conflict.

=> If two persons working on same file then we may get conflicts problem.

=> When conflict occurs we have to resolve those conflicts and we have to commit without conflicts.

Note: When we execute 'git pull' command there is a chance of getting conflicts.

=============
Git Branches
=============

=> Git branches are used to maintain seperate code bases for multiple teams working in the same project.

=> Using git branches multiple teams can work paralelly.

=> In project, one team work shouldn't effect other teams work. We can resolve this issue using git branches concept.

=> In git repo, we will have branches like below

- main (default)
- develop
- feature
- sit
- uat
- release ....

Note: We can use any name for the branch and we can create any number of branches in git repo.

Note: When we use git clone we will get default branch (i.e main)

```
$ git clone <repo-url>

$ git clone -b <branch-name> <repo-url>

# to display current branch name
$ git branch

# To switch branch
$ git checkout <branch-name>
```

What is Pull Request (PR) ?

=> It is used to merge changes from one branch to another branch.

Note: We can select 'Reviewer' to approve our Pull Request.

What is your git repo branching strategy ?

=> Branching strategy represents what are the rules & guidelines we need to follow in git repo for teams colloboration.

Note: Mgmt will decide branching strategy for the project.

main : Final code will be stored here

develop : Ongoing dev activities

feature : Enhancements

SIT : Bug fixing

release : Used for prod release

What is Branch Locking ?

=> When we are deploying code for product there will be code freeze

=> Code Freeze means disabling commit permissions for team members to particular repo/branch.

```
Note: 20 to 30 days brefore prod release, code freeze will happen.

git config
git init
git add
git status
git restore
git commit
git push
git log
git rm
git clone
git pull
git branch
git checkout
git revert

===================
What is git stash
===================

=> It is used to store working tree changes to temp area and make working tree clean.

$ git stash

$ git stash apply

======================
git fetch vs git pull
======================

git pull : directley download changes from central repo to working tree.

git fetch : download central repo changes to local repo.

Note: After git fetch, we need to execute git merge command to merge changes from local repo to
working tree.

        git pull = git fetch + git merge

=========================
git merge vs git rebase
=========================

=> To merge changes from one branch to another branch we will use these commands.

merge : will preseve commit history

rebase : will not preserve commit history

=====================
what is git fork ?
=====================

=> forking means creating copy of other git user repository in our git hub account.

===================
Realtime workflow
===================

1) Management will decide repository server

2) Management will decide branching strategy

3) Development Team should send request to create git repo to DevOps team with manager approval.
```

4) Once git repo created we can create branches based on our requirement.

5) Once our task is completed we need to merge our changes to main branch.

6) When there is production deployment, DevOps team will make code freeze.

Note: If we don't have write permission to git repo then we need to raise request to devops team to get access with manager approval.

Note: If you have any confusion with git operations/branches/branching-strategy take team members help.