SOURCE CODE MANAGEMENT

Lecture 1

# What is GIT?

Git is an open-source version control tool created in 2005 by the developers working on the Linux operating system.

GitHub is a company founded in 2008 that makes tools which integrate with git. You don't need GitHub to use the git but you cannot use GitHub without using git.

Git is used for:

1. Getting code changes.
2. Getting who made the changes.
3. Coding collaborations.

# What is GITHUB?

GitHub is an online hosting service for git repositories. Imagine working a project at home while you are away, maybe at friend's place you suddenly remember the solution to code error that has kept you restless for days.

* Git is not same as GitHub.
* GitHub makes the tool that use git.
* GitHub is the largest host of source code in the world and has been owned by Microsoft since 2018.

# Why do we use Git?

* Over the 70% of developers use git.
* Developers can work together from anywhere in the world.
* Developers can see the full history of project.
* Developers can revert to the earlier versions of the project.

# Version Control System

A version control system tracks the history of changes people had, teams collaborate on projects as developers make the changes to the project, as any earlier versions of the project can be recovered at any time.

## Types of Version Control System:

### Local version control system:

It is stored in your local machine. If the local machine crashes it would not be possible to retrieve the files and all the info will be lost. If anything happens to a single version, all the versions made after will be lost.

### Centralised version control system

In this system there will be a single central server that contains all the files related to the project and many collaborators checkout the files from this single server (you will have only a working copy) the problem with the centralised version control system is if central server crashes almost everything related to the project will be lost.

### Distributed version control system

In this system there will be one or more servers and many collaborators similar to the centralised system but the difference is not only do they check out the latest version but each collaborator will have an exact copy (the rear image) of the main repository on their local machines (including its entire history).

Each user has their own repository and a working copy, this is very useful because even if the server crashes we would not lose everything as several copies are residing in several other computers.

Lecture 2

## What is Repository?

A repository or git project and compasses the entire classes of files and folders associated with a project along with each files revision history.

Repositories in git contains a collection of files of various different versions of a project. These files are important from the repository into the local server of the user for further updations and modifications in the content of the file. The file history appears as snapshots in the time called **commits**. The commits can be organised into multiple lines of development called branches because git is a distributed version control system, repositories are self-contained units and anyone who has a copy of the repository can access the entire code base and its history. Using the command line or other easy of use interfaces a git repository also allows for interaction with the history, cloning the repository, creating branches, committing, merging, comparing changes across versions of code and many more. Platforms like GitHub, git also provide more opportunities for project transparency and collaboration.

Public repositories help teamwork together to build the best possible final product.

You typically obtain a git repository in one of the two ways:

* We can take the local directory that is currently not under the version control and turn it into a git repository.
* We can clone an existing git repository form elsewhere.

In either case we end up with a git repository on our local machine ready for work.

A git repository allows performing various operations on it to create different versions of a project. These operations include the additions of files, creating new repositories, creating an action and deleting a repository. These modifications will result in the creation of different versions of a project.