GIT CLASS-03

GIT BRANCHES:

A branch is a way to isolate development work on a particular aspect of a project.

It is a parallel version of a repository that allows you to work on different features, bug fixes, or experiments without affecting the main codebase.

COMMANDS:

- To see the list of branches : **git branch**
- To create a branch : **git branch branch_name**
- To enter into a branch : **git checkout branch_name**
- To delete a branch : **git branch -d branch_name**
- To delete a branch forcefully : **git branch -D branch_name**
- To rename a branch : **git branch -m old_branch_name new_branch_name**
- To create & switch at a time : **git checkout -b branch_name**

GIT MERGE:

It is used to get the commits from one branch to another

command: **git merge branch_name**

GIT MERGE CONFLICTS:

Merge conflicts happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge.

To resolve the conflicts we need to follow these steps:

- Open the file in VI editor.
- Delete conflict messages.
- Save the file.
- Track the file again and commit it.

To avoid those merge conflicts we need to abort it

command: **git merge --abort**

GIT CHERRY-PICK:

It used to get particular commit from one branch to another

command: **git cherry-pick commit_id**

GIT STASH:

It is used to save your changes but not record them in the Git repository.

COMMANDS:

- To delete the changes temporarily : **git stash**
- To get back the data again : **git stash apply**
- To see the list of stashes : **git stash list**
- To delete all stashes : **git stash clear**
- To delete a particular stash : **git stash drop@{n}**
- To save the stash along with the message : **git stash save "message"**
- To delete first stash : **git stash pop**