**Git Branching:**

**Master Branch:** original code

**Feature Branch**: the master is copied, and new features are added thus creating a feature branch.

**Release Branch:** After all changes and testing the new feature branch is released as release branch. Here the master branch remains the same because the master branch will be used for continuous development. Thus by using the release branch we can deliver the new features to customers.

**Hotfix branch:**

Hotfix branches allow developers to quickly fix critical issues in a production environment by working with the exact code deployed in production. Once the code is fixed, it can be quickly tested and deployed, and later merged into the main code.

Master branch

Release branch created for release

Merged back with master branch

Retained for future development

Delivered to customer

Release Branch

Feature Branch

* Here the first thing is we create a feature branch and do the changes or add the features.
* Next the feature branch is merged with the master branch. Deletes the feature branch.
* Now a release branch is created and after successful testing that will be delivered to the customer.
* If the customer faces issues in the production then the hotfix branch is created and immediate changes are done with testing and merged back with the master branch and released as release branch.

git remote add “repository path”:

This command is used to add the local repository to the remote repository that is to the GITHUB.

We need to paste the git hub repository link or path.

Fork: It is used to create a copy of the repository.

Clone: used to download the code in the repository.

**Branching:**

git checkout -b branchname: this command creates a feature branch with a branch name that has the code up to date.

git branch : gives the branches in the git.

git checkout main: takes us to the main or master branch.

**Merging the feature branch:**

git cherry-pick commit id(in feature branch) : this helps in merging the feature branches with the main branch with using the commit id.

After merging to verify that we can use git log command where the commit in feature is added in the logs of the main branch which was not available previously.

git merge branchname: This is used to merge the feature branch with the main branch. But the commits done in the feature branch will reflect at the top of the commits in the main branch.

git rebase branchname: This is used to merge the feature branch with the main branch. The commits will be in the order of the changes. Helps to track the commits flow.

Sometimes conflicts may occur and we have to remove the conflicts by discussing with developers. After merging again do the add and commit commands.