GIT COMMANDS for Beginners or lone developers

\*\*\*\***CLONE**: Download a copy of a repository (this is usually the first step you take when joining a project to get an overview of the project and where all the files are located)

This function is a one time use

\*\*\*\***ADD**: Add a file to a commit (this is the changes you want to add to the files or commit to the files, it can be a bug fix, feature request, or changes you made to a file, and add those changes to a commit to be later pushed)

One time per commit

\*\*\*\***COMMMIT**: Stage changes with a message (this is a group of changes you made and the message you give to these commits will tell others what changes you have made) you can use ticketing system to label commits to tell others what the commit is intended for

Creates a snapshot/version

\*\*\*\***PUSH**: Copy a commit to repository (this pushes the changes to the repository or cloud so your teammates can pull the changes you made)

\*\*\*\***PULL**: Pull all the commits from the repository

GIT COMMANDS for advance or group developers

**MERGE CONFLICT**

**Check the scenario below where developer A and B made changes to the same file and there is a conflict. Developer B ends up with 3 options, either to merge the changes or discard or overwrite developer A’s changes**

**A screenshot of a computer

Description automatically generated**

**BRANCHES**

**In this case you are working on version updates or long time changes but you still need to be able to make changes to the current version. So you make a new repository called the Branch repository, and after you are done you merge the two repository which will lead to merge conflict. Use the three scenarios above to fix the merge conflict inssue.**

**A screenshot of a computer

Description automatically generated**