Git Report on work flow:

First will create a remote repository.

Command used to create a remote repository **git init –bare /remote.git**

There are 3 developers working for Calculator Project.

The student developers are Bharat, Ravi and Subhash.

Bharat created the base calculator project and pushed that into remote server.

Command: git init in calculator folder.

First all files in com needed to be added to do this used "**git add .**" These are called untacked code files.

And added untracked code files in to local repository using command **"git add . "** Which adds all files in current calculator folder in to staging area. And finally committed the code.

Command: git commit –m "Added Calculator code"

And linked this local repository to remote server repository using command **"git remote add origin /c/Users/Bharat/remote.git"**

And pushed the calculator code using command **"git push origin master"**

And the other developers cloned from the central repository using command.

"git clone file:///c/Users/Bharat/remote.git calculatorSubhash"

"git clone file:///c/Users/Bharat/remote.git calculatorRavi"

Subhash is working in his local repository (calculatorSubhash) and changed Cpu.java and pushed his changes to remote repository. He has done using

git add com/objectsbydesign/mode/Cpu.java . This is used to add the modified code file Cpu.java in to staging area.

git commit –m "subhash changed cpu.java"

git push origin master

Next Ravi is also working in his local repository (calculator) and changed Cpu.java and tried to push his changes to remote repository.

git add com/objectsbydesign/mode/Cpu.java

git commit –m "Ravi changed cpu.java"

git push origin master . This command has returned error because subhash has already pushed changes to remote repository and Ravi's repository got outdated, changes got rejected**.**

So he merged the repository changes in to his local repository using **"git pull origin"** and then pushed his changes to remote repository using "**git push origin master".**

Here I also used another approach "**git fetch origin"** to get repository changes and merged the changes **using "git merge origin" and** then pushed changes to repository using **"git push origin master".**

Bharat to get all other updates:

git pull origin.

Now coming to Branching

And then also I have created a branch in the subhash code

Using git checkout –b feature. This command created feature branch.

In this changed cpu.java and committed the code in branch.

git push -u origin feature. This creates it as a remote tracking branch. After setting up the tracking branch, subhash can call git push without any parameters to push her branch feature. After that to push the changes to remote repository using **git push** which pushes directly to remote repository.

Here the branch changes are pushed to remote repository as this branch is created as remote tracking branch.

And in Ravi's code done following steps to get subhash code of branch.

Checking the information of remote repository.
git remote show origin
Fetching the latest data from the repository
git fetch origin
I am tracking the branch origin/feature
git checkout -b ravifeature origin/feature
Now I can see the subhash branch code and modifications as got feature branch from subhash here.
Now I am merging the subhash code
git checkout master
git merge ravifeature. This automatically merges subhash feature branch code in to ravi repository.
Now subhash code is merged in to ravi calculator project.

Bharat also created branch

Git checkout –b bharatbranch

Staged cpu.java

Git add com/objectsbydesign/calc/model/cpu.java

Git commit –m "Bharatbranch"

Git push origin. bharatbranch pushed to origin.

Like this 3 developers can work on their local repositories and sync with remote repository when they complete the development features.