<p style="text-align:center;color:red">GIT</p>

<p style="color:red">Download</p>

Google--->search ' Git Download'-->click First Link--->Download for Windows

<p style="color:red">To Create a repository</p>

After Login-->click "+" near user profile-->select new repository--->add repository name-->click public(default)-->click create repository(nothing to select anymore)

<p style="color:red">Commands for PUSH</p>

In local machine(laptop) create a empty folder-->double click the folder-->right click-->select 'Git bash here'-->git command prompt is open-->

<p style="color:red">Commands:</p>

<mark>For first time push the code to Master (lead)</mark>

1. git config --global user.name 'git account user name'

2. git config --global user.email 'user email-id'

3. git init (to initialize the project-->without init nothing can be done-->after this .git hidden folder will be created in above mentioned empty folder)
4. git status(to check whether the project is ready to move on stagging area)(here comes the 'red' or 'green' color notification will appear)

5. git add . (to add all files)

   5.1. git add 'specific file name'(to add specific file)

6. git commit -m "message"(to make commitment between stagging area and remote machine)

7. git remote add origin 'repo url'(master-repo url)

8. git push -u origin master

   8.1. git push -u origin 'branch name'(if we move the code to concern branch)

Create a branch from master

In git hub-->click Repository name-->click master-->type a branch name-->click create branch from master (branch will be created)

In branch copy of the master code will be present

In local machine(laptop) create a empty folder-->double click the folder-->right click-->select 'Git bash here'-->git command prompt is open-->

commands:

1. git clone 'repo url' (branch-repo url (code))

now project folder will be in above created empty folder..

To transfer the above project folder to Eclipse

----

1. Go to Eclipse-->File-->select switch Workspace-->others--browse-->select Project folder-->click ok-->click Launch(new workspace will be created)

2. In new workspace-->File-->Import-->select General-->Existing Projects into workspace-->browse-->select the Project Folder-->click ok-->click Finish..

(In workspace new Project will be created in master branch name)

To change from Master to our own branch

In eclipse Right click Master Project-->properties-->location-->click show in system explorer icon-->navigate project folder-->doulbe click-->right click-->

select Git Bash here-->coomand prompt will open

Commands:

git checkout 'branch name'(our own branch name)-->(now project name is changed from master to our own branch name in Eclipse)now you can work on the project..
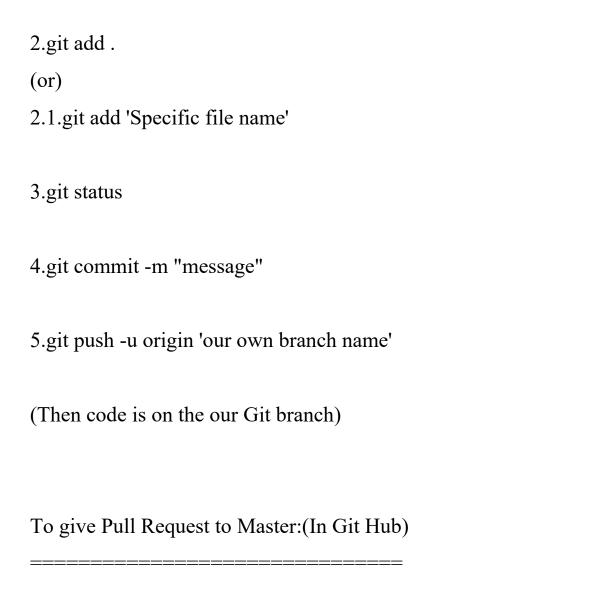
To PUSH updated code of our branch from Eclipse to Remote Branch(GitHub)

In eclipse Right click our own branch Project-->properties-->location-->click show in system explorer icon-->navigate project folder(in our Laptop)-->

doulbe click-->right click-->select Git Bash here-->command prompt will open

Commands:

1.git status

2.git add .

(or)

2.1.git add 'Specific file name'

3.git status

4.git commit -m "message"

5.git push -u origin 'our own branch name'

(Then code is on the our Git branch)

To give Pull Request to Master:(In Git Hub)

=================================

Select our own branch-->click pull request-->click compare and pull request-->then message show as 'Able to merge'-->write a comment in comment box-->

click Create PULL request-->It shows 'This branch has no conflicts with the base branch'-->click Merge PULL request-->click Confirm Merge(Lead Work)-->
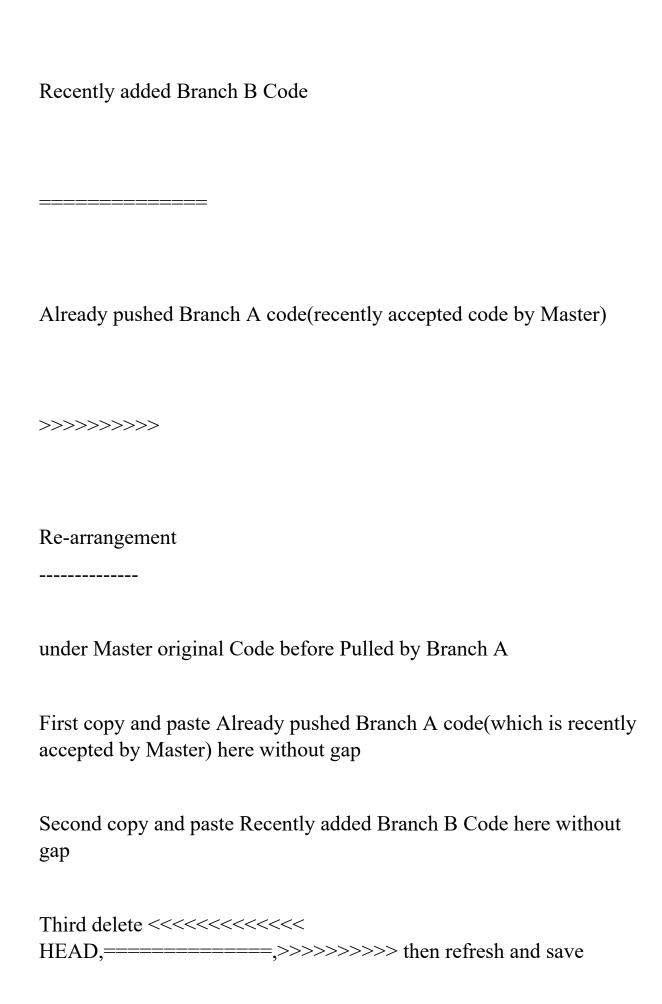
Then Lead will give the message in comment Box about our Merge request..If accepted Success Message will be as "Pull request succefully merged and closed"..

Conflict:

========

Branch A from Master-->Clone the code from Master

Branch B from Master-->Clone the code from Master

Branch A from Master

1.Get the Code from Master and work on that code and PUSH the code to A Branch and give PULL request and If accepted code will be merged in Master...

Now Master has new updated code...

Branch B from Master

Here Branch B not aware of the above PULL request he works on the old code which is get from Master already..Branch B work on that code and PUSH the code to B Branch and give PULL request-->Click Compare and PULL request-->Now Conflict taken place with a message 'can't automatically merge'-->In comment write a message for PULL request-->click PULL request..Now it shows the Conflict message as 'This branch has Conflicts that must be resolved'..
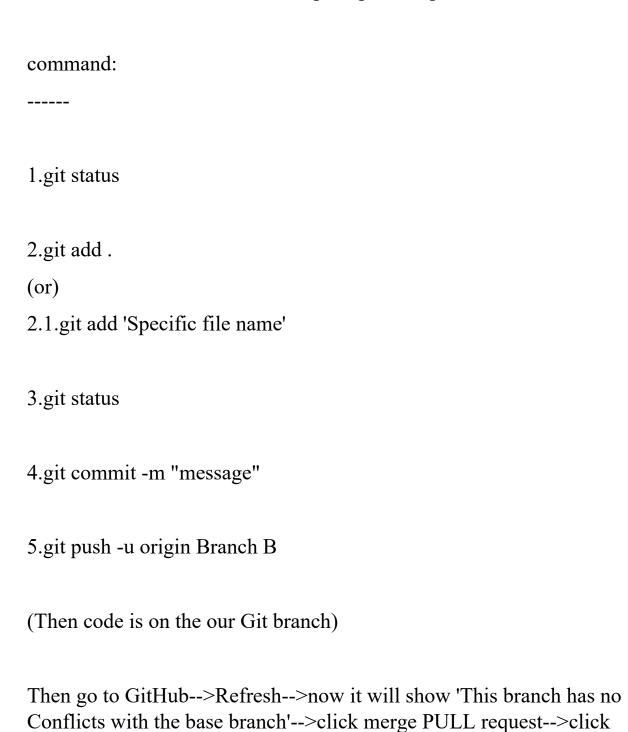
Steps to resolve the Conflict

1.Go to Eclipse which recently worked Branh B workspace-->Open our Branch B-->Now it shows in that project as 'Conflicts B Branch'..

2.Now Right click Conflicted Project-->properties-->location-->click show in system explorer icon-->navigate project folder-->doulbe click-->right click-->

select Git Bash here-->command prompt will open

Commands

----

git pull origin master-->Enter

Come back to our Branch B workspace in Eclipse-->open the Project-->now it shows

Master original Code before Pulled by Branch A

<<<<<<<<<<<<

Recently added Branch B Code

===============

Already pushed Branch A code(recently accepted code by Master)

>>>>>>>>>>

Re-arrangement

----

under Master original Code before Pulled by Branch A

First copy and paste Already pushed Branch A code(which is recently accepted by Master) here without gap

Second copy and paste Recently added Branch B Code here without gap

Third delete <<<<<<<<<<<<<<
HEAD,===============,>>>>>>>>>> then refresh and save

Then Right click Conflicted Project-->properties-->location-->click show in system explorer icon-->navigate project folder-->doulbe click-->right click-->

select Git Bash here-->command prompt will open

command:

1.git status

2.git add .
(or)
2.1.git add 'Specific file name'

3.git status

4.git commit -m "message"

5.git push -u origin Branch B

(Then code is on the our Git branch)

Then go to GitHub-->Refresh-->now it will show 'This branch has no Conflicts with the base branch'-->click merge PULL request-->click confirm merge(lead)-->

If code accepted it shows a message 'pull request successfully merged and closed' with command from our Lead...