Install GIT & make sure it is added into PATH

```
# To check version of git
$ git --version
git version 2.31.1.windows.1
```

A1 solutions (Use GIT as local VCS):

---

1. Create a directory ‘project_dir' & cd to 'project_dir’

   ```
   # To create new directory
   $ mkdir project_dir

   # To change directory path
   $ cd project_dir
   ```

2. Initialize git version database

   ```
   $ git init
   Initialized empty Git repository in C:/Users/LENOVO/project_dir/.git/
   ```

3. Create a new file index.html

   ```
   $ touch index.html
   ```

4. Check the git status. You should find index.html as an untracked file.

   ```
   $ git status
   On branch master

   No commits yet

   Untracked files:
     (use "git add <file>..." to include in what will be committed)
         index.html

   nothing added to commit but untracked files present (use "git add" to track)
   ```

5. Stage the index.html file.

   ```
   $ git add index.html
   ```

6. Commit index.html

```
$ git commit -m 'added index.html'
[master (root-commit) b16a665] added index.html
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

7. Make few changes in index.html & create a new file info.txt file.

```
# Changes in index.html
<html>
        <head><title>section-0</title></head>
        <body>
        <p>Created index.html file.</p>
        </body>
</html>

# To create info.txt
$ touch info.txt
```

8. Check git status. You should find index.html & info.txt as untracked files

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
      modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      info.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

But here I get index.html as a modified file because we did the changes in the file and it’s into the staging area.

9. Configure GIT to ignore all txt files

```
$ touch .gitignore
/info.txt
```

10. Again check the git status. You should find only index.html as an untracked files.

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
```

But here I get index.html as a modified file and .gitignore as an untracked file

11. State & commit index.html

```
# Adding to staging area
$ git add index.html

# Committing index.html file
$ git commit -m 'added code in index.html'
[master 95be17f] added code in index.html
 1 file changed, 6 insertions(+)
```

12. Log all your comments so far

```
$ git log
commit 95be17f104717710f701b4e34e584a5e0632891c (HEAD -> master)
Author: Omkar <javomkar@gmail.com>
Date:   Mon May 17 18:53:09 2021 +0530

    added code in index.html
commit b16a66537beaf9001cee0e41e0cbbcf1f6d8570e
Author: Omkar <javomkar@gmail.com>
Date:   Mon May 17 18:00:23 2021 +0530

    added index.html
```

13. Make some changes in index.html

```
# Updated code of index.html
```

```
<html>
        <head><title>section-0</title></head>
        <body>
        <p>Created index.html file.</p>
        <p>Added new line into index.html file</p>
        </body>
</html>
```

14. Revert the change made in the previous step using git command

```
$ git checkout -- index.html
<html>
        <head><title>section-0</title></head>
        <body>
        <p>Created index.html file.</p>
        </body>
</html>
```

15. Again change index.html

```
# Added updated code to index.html
<html>
        <head><title>section-0</title></head>
        <body>
        <p>Created index.html file.</p>
        <p><b>Added new line into index.html file</b></p>
        </body>
</html>
```

16. Stage index.html

```
$ git add index.html
```

17. Revert back the last stage.

```
$ git rm --cached index.html
rm 'index.html'
```

18. Rename 'add’ command to 'my-add’

```
$ git config --global alias.my-add add
```

19. Using my-add command Stage index.html again & commit the changes

```
# Adding to staging area
$ git my-add index.html

# To commit the changes
$ git commit -m 'Made paragraph bold'
[master 954b6e4] Made paragraph bold
 1 file changed, 7 insertions(+)
```

20. Revert the last commit

```
# To edit the commits
$ git rebase -i HEAD~2
interactive rebase in progress; onto b16a665
Last commands done (2 commands done):
   drop 954b6e4 Made paragraph bold
   pick 4eeee1d This reverts commit
954b6e4b204b81dc031a47605d35c446d85d7a5b.
No commands remaining.
You are currently rebasing branch 'master' on 'b16a665'.
  (all conflicts fixed: run "git rebase --continue")

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
The previous cherry-pick is now empty, possibly due to conflict resolution.
If you wish to commit it anyway, use:

    git commit --allow-empty

Otherwise, please use 'git rebase --skip'
Could not apply 4eeee1d... This reverts commit
954b6e4b204b81dc031a47605d35c446d85d7a5b.

# To revert the commit
$ git commit --allow-empty
[detached HEAD 5fccc38] This reverts commit
954b6e4b204b81dc031a47605d35c446d85d7a5b.
 Date: Mon May 17 19:45:04 2021 +0530

# To get back to master branch
$ git rebase --skip
Successfully rebased and updated refs/heads/master.
```

A2 solutions (GIT Branching)

---

21. First take a backup of your assignments & projects. This is required because due to incorrect GIT operation you may lose your files

22. Create an empty directory 'Assignments' & cd to 'Assignments'

    ```
    # To create new directory
    $ mkdir Assignments

    # To change path of directory
    $ cd Assignments

    # To initialize directory
    $ git init
    Initialized empty Git repository in C:/Users/LENOVO/Assignments/.git/
    ```

23. Create a file README.txt inside 'Assignments' & write few lines about the contents of 'Assignments' folder

    ```
    # Created README.txt file
    $ touch README.txt

    # Content of README.txt file
    This is README.txt file
    ```

24. Commit README.txt file.

    ```
    # Adding to staging area
    $ git add README.txt

    # Committed README.txt
    $ git commit -m "added README.txt"
    [master (root-commit) 3263ec6] added README.txt
     1 file changed, 1 insertion(+)
     create mode 100644 README.txt
    ```

25. Now create a new branch 'html-assignments'

    ```
    $ git branch html-assignments
    ```

26. Switch to 'html-assignments' branch

```
$ git checkout html-assignments
Switched to branch 'html-assignments'
```

27. Copy all HTML assignments inside 'Assignments' folder

28. Commit HTML assignments into 'html-assignments' branch

```
$ git commit -m 'added index.html'
[html-assignments a51faac] added index.html
 1 file changed, 12 insertions(+)
 create mode 100644 index.html
```

29. Make minor changes into few files belonging to 'html-assignments' branch

30. Commit those changed files

```
# Adding to staging area
$ git add index.html

# Committed new comment
$ git commit -m 'updated title of index.html'
[html-assignments 066cbf7] updated title of index.html
 1 file changed, 1 insertion(+), 1 deletion(-)
```

31. Switch to master branch

```
$ git checkout master
Switched to branch 'master'
```

32. Make minor changes into README.txt file & commit those changes into master

```
# Adding to staging area
$ git add README.txt

# Committed new comment
$ git commit -m 'updated README.txt'
[master 8d3e95c] updated README.txt
 1 file changed, 1 insertion(+), 1 deletion(-)
```

33. Again switch to 'html-assignments' branch.

    $ git checkout html-assignments
    Switched to branch 'html-assignments’

34. Make minor changes into a few files belonging to 'html-assignments' branch.

35. Commit those changes

    # Adding to staging area
    $ git add index.html

    # Committed new comment
    $ git commit -m 'changed paragraph to italic'
    [html-assignments 11029f5] changed paragraph to italic
     1 file changed, 1 insertion(+), 1 deletion(-)

36. Switch to master

    $ git checkout master
    Switched to branch 'master'

37. Merge 'html-assignments' branch into master. Confirm all html assignments are shown in master

    $ git merge master html-assignments
    Merge made by the 'recursive' strategy.
     index.html | 12 ++++++++++++
     1 file changed, 12 insertions(+)
     create mode 100644 index.html

38. Finally delete the 'html-assignments' branch

    $ git branch -d html-assignments
    Deleted branch html-assignments (was 11029f5).

CSS assignments:

---

1. Create a new branch 'css-assignments'
   $ git branch css-assignments

2. Switch to the 'css-assignments' branch

   ```
   $ git checkout css-assignments
   Switched to branch 'css-assignments'
   ```

3. Copy all CSS assignments inside 'Assignments' folder

4. Commit CSS assignments into 'css-assignments' branch

   ```
   # Adding to staging area
   $ git add style.css

   # Committed style.css file
   $ git commit -m 'added style.css file'
   [css-assignments c7e4e02] added style.css file
    1 file changed, 39 insertions(+)
    create mode 100644 style.css
   ```

5. Make minor changes into README.txt file on line 1 belonging to 'css-assignments' branch

6. Commit those changed files

   ```
   $ git add README.txt

   $ git commit -m 'added css line to README.txt'
   [css-assignments 7e3df60] added css line to README.txt
    1 file changed, 1 insertion(+)
   ```

7. Switch to master branch

   ```
   $ git checkout master
   Switched to branch 'master'
   ```

8. Make minor changes into README.txt file on line 3 & commit those changes into master

   ```
   $ git add README.txt

   $ git commit -m 'updated 3rd line'
   [master cf14286] updated 3rd line
    1 file changed, 2 insertions(+)
   ```

9. Again switch to 'css-assignments' branch

   $ git checkout css-assignments
   Switched to branch 'css-assignments'

10. Make minor changes into few files belonging to 'css-assignments' branch

11. Commit those changes

    $ git add .
    $ git commit -m 'deleted document type from style.css'
    [css-assignments e7538aa] deleted document type from style.css
     1 file changed, 1 deletion(-)

12. Switch to master

    $ git checkout master
    Switched to branch 'master'

13. Merge 'css-assignments' branch into master. Confirm all css assignments are shown in master

    $ git merge master css-assignments
    Auto-merging README.txt
    CONFLICT (content): Merge conflict in README.txt
    Automatic merge failed; fix conflicts and then commit the result.

    $ git commit -a
    [master 7c69006] Merge branch 'css-assignments'

    $ git merge master css-assignments
    Already up to date.

14. Finally delete the 'css-assignments' branch

    $ git branch -d css-assignments
    Deleted branch css-assignments (was e7538aa).

JavaScript Assignments

---

1. Create a new branch ’js-assignments’

   $ git branch js-assignments

2. Switch to js-assignments’ branch

   $ git checkout js-assignments
   Switched to branch 'js-assignments'

3. Copy all JavaScript assignments inside the 'js-assignments' folder.

4. Commit JavaScript assignments into the ‘js-assignments’ branch.

   $ git add hello.js

   $ git commit -m 'added hello.js'
   [js-assignments 83ed2c4] added hello.js
   1 file changed, 2 insertions(+)
   create mode 100644 hello.js

5. Make minor changes into README.txt file on line 1 belonging to ‘js-assignments’ branch

6. Commit those changed files

   $ git add README.txt

   $ git commit -m 'added js line'
   [js-assignments 1080860] added js line
   1 file changed, 2 insertions(+), 4 deletions(-)

7. Switch to master branch

   $ git checkout master
   Switched to branch 'master'

8. Make minor changes into README.txt file on line 1 & commit those changes into master

   $ git add README.txt
   $ git commit -m 'updated 1st line'
   [master b873fef] updated 1st line
    1 file changed, 4 insertions(+), 6 deletions(-)

9. Again switch to js-assignments branch

   $ git checkout js-assignments
   Switched to branch 'js-assignments'

10. Make Minor changes into few files belonging to js-assignments branch

11. Commit those changes

    $ git add .

    $ git commit -m 'updated all files'
    [js-assignments d9a7beb] updated all files
     1 file changed, 1 insertion(+), 1 deletion(-)

12. Switch to master

    $ git checkout master
    Switched to branch 'master'

13. Merge ‘js-assignments’ branch into master. Confirm all javascript assignments are shown in master.

    $ git merge master js-assignments
    Auto-merging README.txt
    CONFLICT (content): Merge conflict in README.txt
    Automatic merge failed; fix conflicts and then commit the result.

    $ git add .

    $ git commit -m 'fixed conflicts'
    [master 3f64ced] fixed conflicts

    $ git branch --merged
      js-assignments
    * master

14. Finally delete js-assignments branch

    $ git branch -d js-assignments
    Deleted branch js-assignments (was d9a7beb).

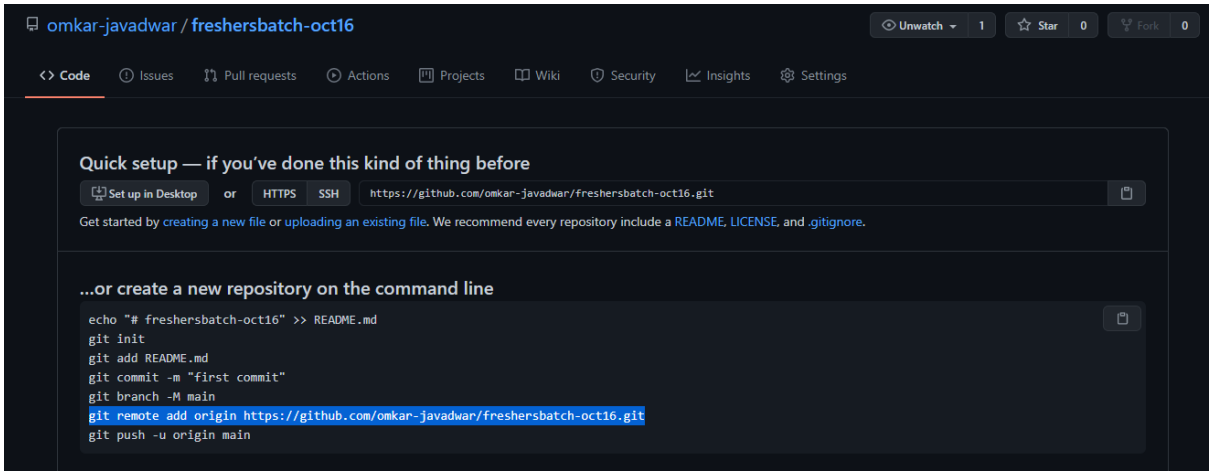
A3 solutions (GIT Remoting)

Pushing assignments to remote repository

39. Create a github account if you do not have already.

40. Login into the github account.

41. Create a new public repository 'freshersbatch-oct16’.



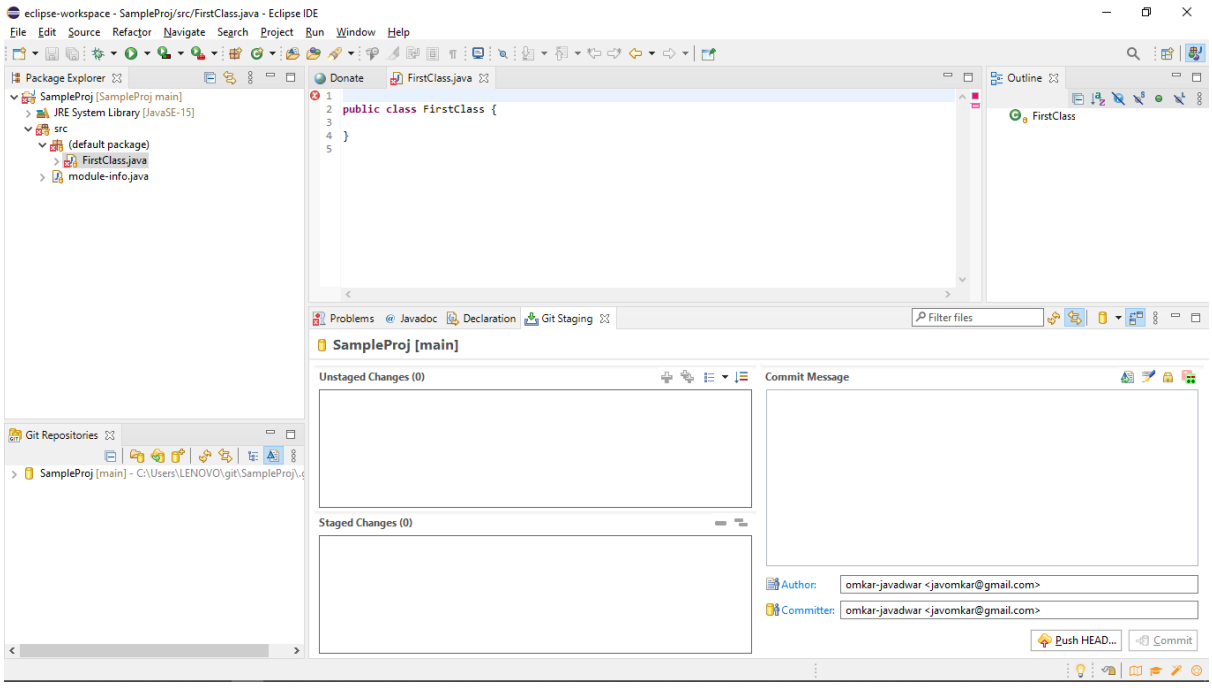
42. Commit & push any sample file to this repository under 'Assignments' directory.

https://github.com/omkar-javadwar/freshersbatch-oct16

A4 solutions(Pushing source code to remote repository using Eclipse GIT plugin)

---

1. One developer from the project team will create eclipse projects 'SampleProj' & add sample source code files. Then commit all files through eclipse GIT plugin.



2. Collaborate other team members with your github account so that they can also modify the committed files.

3. Other developers from the same team will checkout all files from the remote repository. This might get conflicts since certain files fail to merge. In such a case, merge it manually.

4. Commit & push the 'SampleProj' project.

   https://github.com/omkar-javadwar/SampleProj