

Git Assignment Objective: The goal of this assignment is to help you get familiar with basic Git commands and workflows, which are commonly used in real-world software development.

### 1. Create a New Repository. (done -github repo)

`Mkdir NUCLEUSTEQ_AKSHAT`

`Cd GIT_Class`

`git init`

### 2. Create a New File (done)

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> echo "Adding new commands -> git blanch new_commands ->git checkout new_commands(to switch to new branch) -> now add new data to file" >> assignment1_updated.txt
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class>
```

### 3. Create a Branch (done)

```
Create mode 100644 GIT_Class/assignment1_updated.txt
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git branch updated_new
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git checkout updated_new
Switched to branch 'updated new'
```

### 4. Make Changes in the New Branch(done)

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git checkout updated_new
Switched to branch 'updated new'
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> echo "Helloo ji" >> assignment1_updated.txt
```

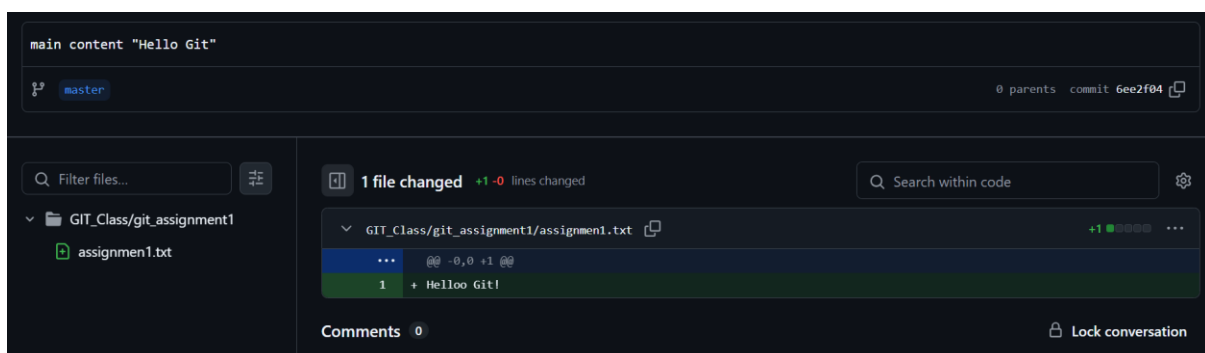
### 5. Create Another Branch (Optional)(done IN ASSIGNMENT1.TXT NEW\_FEATURES AND NEW\_COMMANDS AS BRANCH )

### 6. Merge Branches(done)

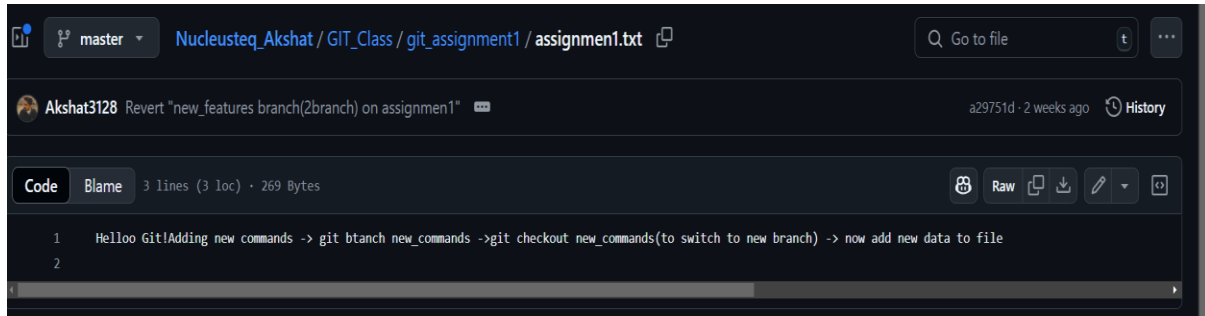
Merge done on new file assignment1\_updated with no merge conflict.

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git merge updated_new
Updating a8cb928..d6e9e5a
Fast-forward
 GIT_Class/assignment1_updated.txt | Bin 260 -> 282 bytes
 1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git add .
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git commit -m "New branch merged on assignment1_updated"
[master e5d4d0f] New branch merged on assignment1_updated
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 GIT_Class/git assignment1/~$t Assignment Objective.docx
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class>
```

Merge conflict removed for assignment1.txt having branch new\_commands and new\_features .It was basically due to an old file being deleted during using branch deletion,del file and revert commands in the next steps.Before there were no merge conflict when merging new\_features on master as can be seen from history.I have attached screenshot of instances for better reference.



This screenshot shows a GitHub commit view for the file `assignment1.txt`. The commit message is "main content 'Hello Git'" and it is on the `master` branch. The commit hash is `6ee2f04`. The file content is shown as a diff, indicating a change from an empty file to one containing "Hello Git!". The interface includes a search bar, a file explorer on the left, and a "Lock conversation" button at the bottom right.



This screenshot shows a GitHub commit view for the file `assignment1.txt`. The commit message is "Revert 'new\_features branch(2branch) on assignment1'" and it is on the `master` branch. The commit hash is `a29751d`. The file content is shown as a diff, indicating a change from an empty file to one containing "Hello Git!". The interface includes a search bar, a file explorer on the left, and a "Lock conversation" button at the bottom right.

Below is the resolution of merge conflict which happened after deleting one file due to testing and after using revert.

```
Switched to branch 'new_features'
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge new_features
CONFLICT (modify/delete): GIT_Class/git_assignment1/assignment1.txt deleted in HEAD and modified in new_features. Version new_features of GIT_Class/git_assignment1/assignment1.txt left in tree.
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout master
GIT_Class/git_assignment1/assignment1.txt: needs merge
error: you need to resolve your current index first
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git status
On branch master
Your branch is up to date with 'origin/master'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add/rm <file>..." as appropriate to mark resolution)
    deleted by us:   GIT_Class/git_assignment1/assignment1.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout new_features
GIT_Class/git_assignment1/assignment1.txt: needs merge
error: you need to resolve your current index first
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git add GIT_Class/git_assignment1/assignment1.txt
>>
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git commit -m "Resolved merge conflict in assignment1.txt"
>>
[master 8284754] Resolved merge conflict in assignment1.txt
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge new_features
Already up to date.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
```

```
nothing to commit, working tree clean
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout new_commands
Switched to branch 'new_commands'
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge new_commands
Already up to date.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)







nothing to commit, working tree clean
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout new_feature
Switched to branch 'new_feature'
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge new_feature
warning: Cannot merge binary files: GIT_Class/git_assignment1/assignment1.txt (HEAD vs. new_feature)
Auto-merging GIT_Class/git_assignment1/assignment1.txt
CONFLICT (content): Merge conflict in GIT_Class/git_assignment1/assignment1.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git abort
git: 'abort' is not a git command. See 'git --help'.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge --abort
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git branch -D new_feature
Deleted branch new_feature (was 9d3e30d).
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)

Add Logs  🚀 CyberCoder  Improve Code  Share Code Link  Generate Commit Message  LF  {} Binary  🌐 Go Live  ⚡ AI Code Chat  CO

Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge new_feature
warning: Cannot merge binary files: GIT_Class/git_assignment1/assignment1.txt (HEAD vs. new_feature)
Auto-merging GIT_Class/git_assignment1/assignment1.txt
CONFLICT (content): Merge conflict in GIT_Class/git_assignment1/assignment1.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git abort
git: 'abort' is not a git command. See 'git --help'.
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git merge --abort
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git branch -D new_feature
Deleted branch new_feature (was 9d3e30d).
PS C:\Users\Acer\Desktop\Nucleus_Akshat> git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)



nothing to commit, working tree clean
PS C:\Users\Acer\Desktop\Nucleus_Akshat> 
```

Below is the history for the changes done :

added updated_assignment1 to assignment folder Akshat3128 committed 23 minutes ago	000a77e		<>
WORD DOC Updated Akshat3128 committed 26 minutes ago	6ba53a4		<>
New branch merged on assignment1_updated Akshat3128 committed 42 minutes ago	e5d4d0f		<>
New feature added--hello ji on branch updated_new for practice Akshat3128 committed 45 minutes ago	d6e9e5a		<>
New file created for gitassignment1 named gitassignment1_updated Akshat3128 committed 51 minutes ago	a8cb928		<>
Resolved merge conflict in assignment1.txt Akshat3128 committed 1 hour ago	8284754		<>

## 7. Pull Changes from Remote(done as well as push)

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git pull origin master
From https://github.com/Akshat3128/Nucleusteq_Akshat
* branch                master      -> FETCH_HEAD
Already up to date.
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class>
```

Logs  CyberCoder Improve Code Share Code Link Generate Commit Message LF {} Binary 

## 8. View Git Log and History

Used git log --oneline for crisp output in a single line.

Used git --all to see all history from others branches, not just main branch.

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git log --oneline
6d47537 (HEAD -> master, origin/master, origin/HEAD) delete assignment1.txt
5ccb4d5 (new_features) new_features branch(2branch) on assignmen1
ceafba5 (new_commands) new_commands branch on assignmen1
908731c updated_assignment1.txt
6ee2f04 main content "Hello Git"
```

```

PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git log --all
commit 5ccdd4795ea9ce428e6a7d4c154348e864bdda96 (new_features)
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:41:34 2025 +0530

    adding something new to new feature branch

commit 7ec0e990ab9019cbd1dc5995e02194d98c94aa69 (origin/master, origin/HEAD)
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:33:35 2025 +0530

    Revert "name corrected for assignmen1"

    This reverts commit 1c0d0b5984dcad4e0048d62d899dec0809f2c03f.

commit 1c0d0b5984dcad4e0048d62d899dec0809f2c03f (HEAD -> master)
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:30:19 2025 +0530

    name corrected for assignmen1

commit a29751dae08ee618d2c4bb1f2733d3e897bf6aa2
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:10:30 2025 +0530

    Revert "new_features branch(2branch) on assignmen1"

    This reverts commit 5ccb4d5417bff0e44ffb56ce38e74caee65b3dad.

    used git revert to revert the commit to sha id 5ccb4d5

```

```

PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git log
commit 1c0d0b5984dcad4e0048d62d899dec0809f2c03f (HEAD -> master)
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:30:19 2025 +0530

    name corrected for assignmen1

commit a29751dae08ee618d2c4bb1f2733d3e897bf6aa2
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:10:30 2025 +0530

    Revert "new_features branch(2branch) on assignmen1"

:....skipping...
commit 1c0d0b5984dcad4e0048d62d899dec0809f2c03f (HEAD -> master)
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:30:19 2025 +0530

    name corrected for assignmen1

commit a29751dae08ee618d2c4bb1f2733d3e897bf6aa2
Author: Akshat3128 <akshatsaxena62941@gmail.com>
Date: Tue Feb 18 20:10:30 2025 +0530

    Revert "new_features branch(2branch) on assignmen1"

    This reverts commit 5ccb4d5417bff0e44ffb56ce38e74caee65b3dad.

    used git revert to revert the commit to sha id 5ccb4d5

:....skipping...
commit 1c0d0b5984dcad4e0048d62d899dec0809f2c03f (HEAD -> master)

```

## 9. Git Status and Staging

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git status
On branch master
Your branch is up to date with 'origin/master'.
```

Added assignment1\_updated to Staging Area:

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        assignment1_updated.txt
        git_assignment1/~$ Assignment Objective.docx

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git add assignment1_updated.txt
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   assignment1_updated.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        git_assignment1/~$ Assignment Objective.docx

PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class>
```

File committed hence removed from staging area:

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class> git commit -m "New file created for gitassignment1 named gitassignment1_updated"
[master a8cb928] New file created for gitassignment1 named gitassignment1_updated
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 GIT_Class/assignment1_updated.txt
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class>
```

## 10. Revert and Reset

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git revert 5ccb4d5
[master a29751d] Revert "new features branch(2branch) on assignment1"
1 file changed, 0 insertions(+), 0 deletions(-)
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1>
```

```
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git reset --soft HEAD~1
PS C:\Users\Acer\Desktop\Nucleus_Akshat\GIT_Class\git_assignment1> git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    assignment1.txt
```

used git reset --soft to undo the last commit while keeping changes --hard can be used to remove changes which can be seen as my local repo is lagging the master repo by 1 commit