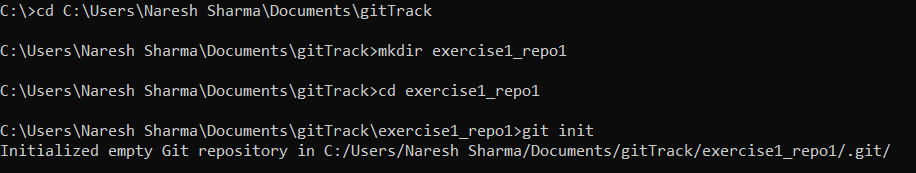
**GIT EXRECISE**

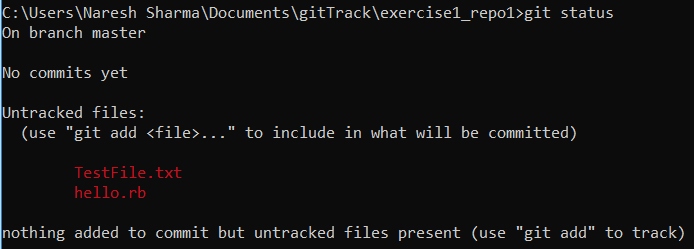
|  |  |
| --- | --- |
| **Action** | **Command Used** |
| Create Repository | git init |
| Check status | git status |
| Add untracked files to staging area | git add . |
| Commit | git commit -m “message” |
| Add files to staging area and commit in one go | git commit -am “message” |
| Remove file | git rm <file-name> |
| Add .gitignore fle | touch .gitignore |
| Check difference | git diff <file-name> or git diff head |
| Revert untracked changes | git checkout <file-name> |
| Revert tracked changes | git reset |
| Revert committed changes | git reset --hard <commit-hash> |
| Display all commits | git log |
| Display specific commit | git show <commit-hash> |
| Create branch | git checkout -b <branch-name> |
| Delete branch | git branch -d <branch-name> |
| Merge branch | git merge <branch-name> |
| Rebase branch | git rebase <branch-name> |
| Stash changes | git stash |
| Retrieve stashed changes | git stash apply |
| Add local repository to github | git remote add origin <github-repo-url> |
| Push chnages to remote | git push -u origin <branch- names> |
| List all branches(local + remote) | git branch -a |
| List all remote branches | git branch -r |

1. Create a repository

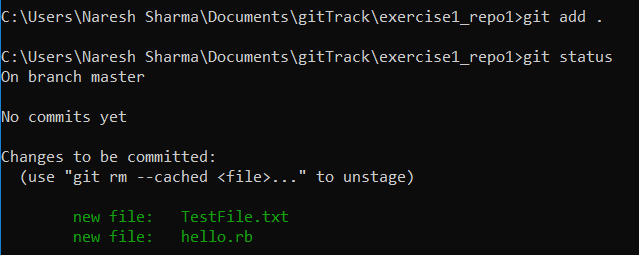


1. Make some changes -- add some files, change content of some files

Added two untracked files.

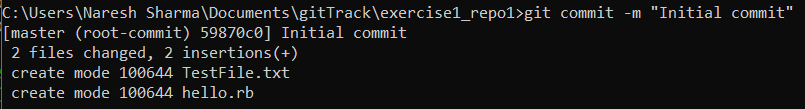


Added these untracked files to staging area using command git add .

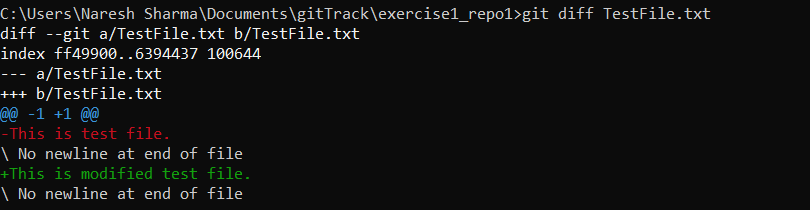


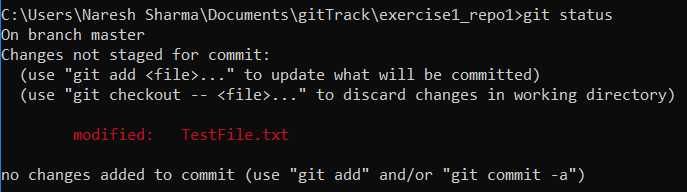
1. commit changes

Committed changes to repository using git commit -m “Initial commit”



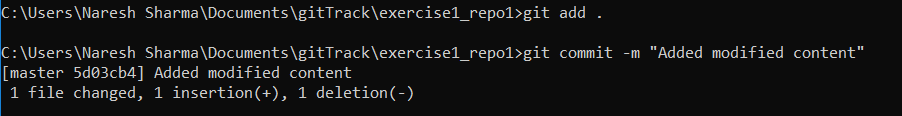
Again modified TestFile.txt and checked the diff using git diff TestFile.txt





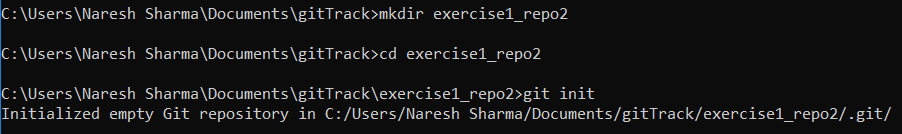
commit changes

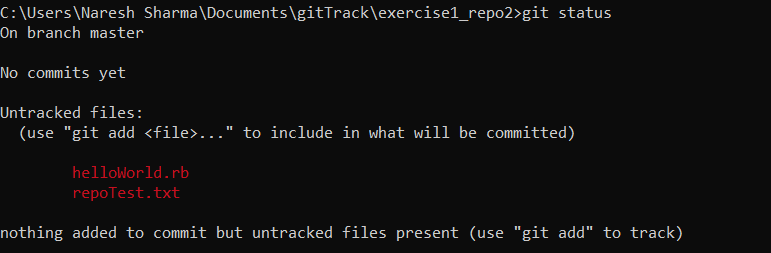
Added modified file to staging area and the committed.

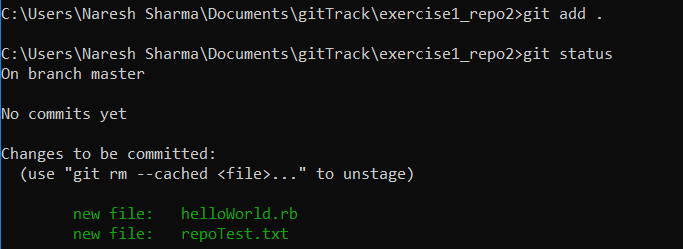


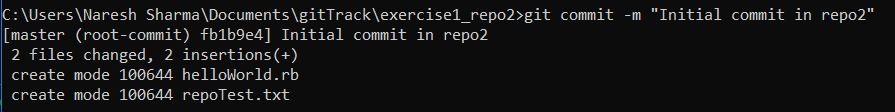
1. repeat steps 1-3 for 2 more times. try to remove and ignore some files . Use git diff before each commit.

Repeated steps 1-3 first time

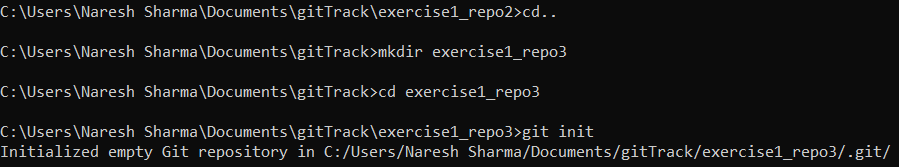


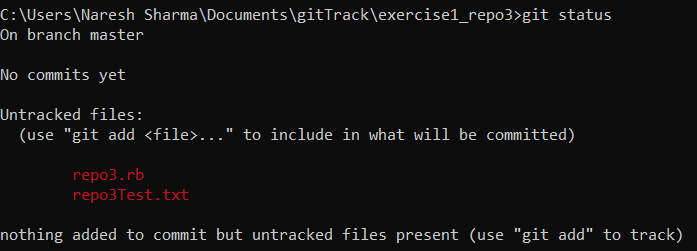


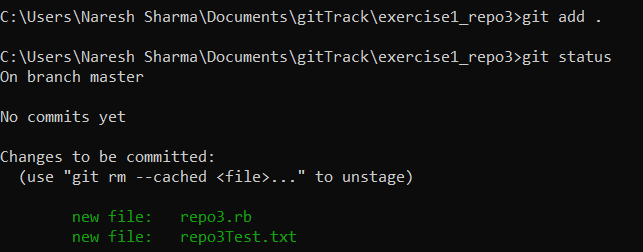


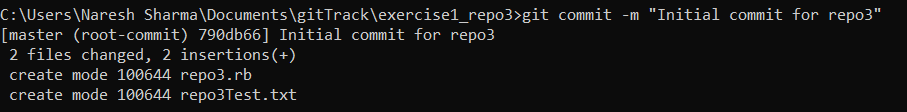


Repeated steps 1-3 second time



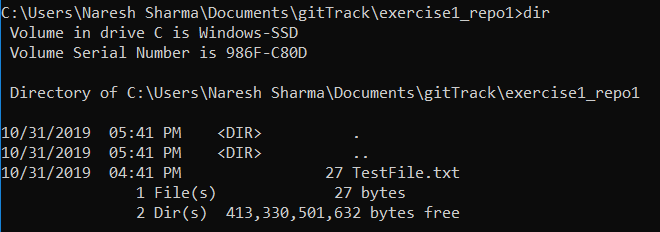


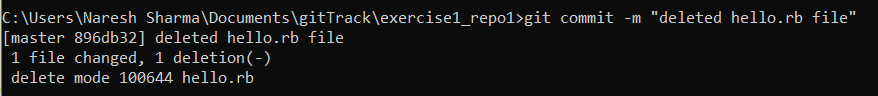




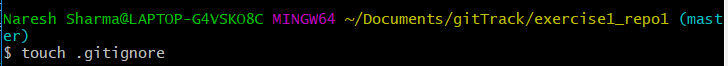
Remove File from git repository using git rm hello.rb



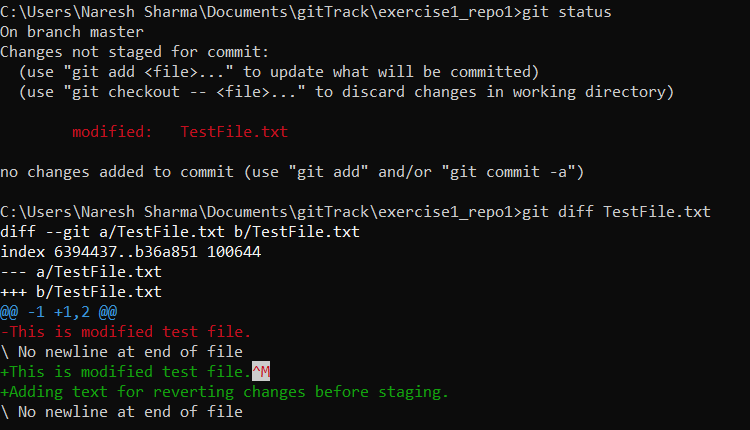




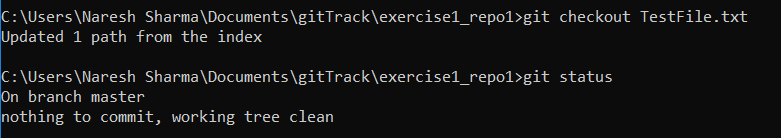
Added .gitignore file to the repository using touch .gitignore



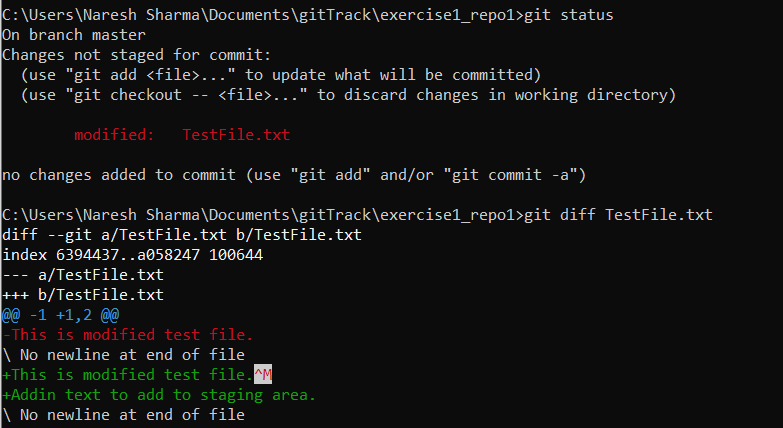
1. create changes and revert those changes using git when those changes are not added to staging are

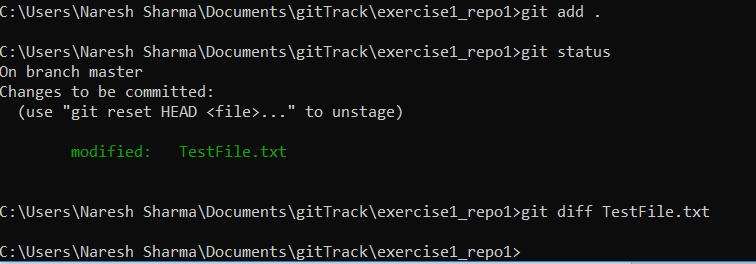


Command used :- git checkout TestFile.txt

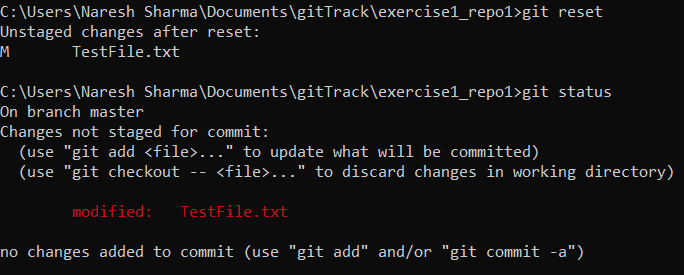


1. create changes and revert those changes after adding changes to staging area

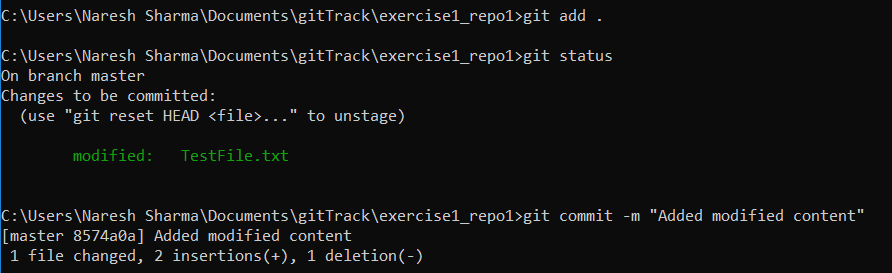


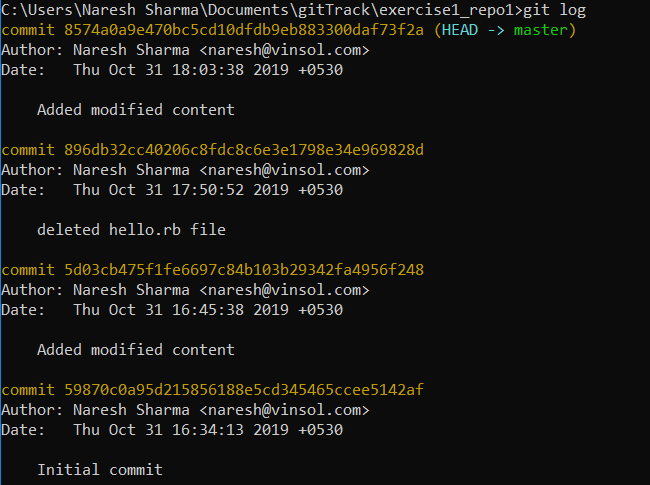


Command Used :- git reset

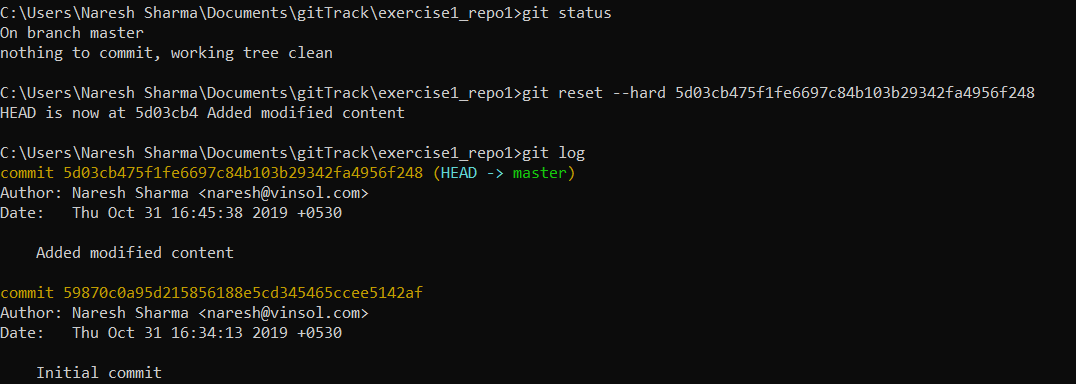


1. create change and revert those changes after committing those changes .



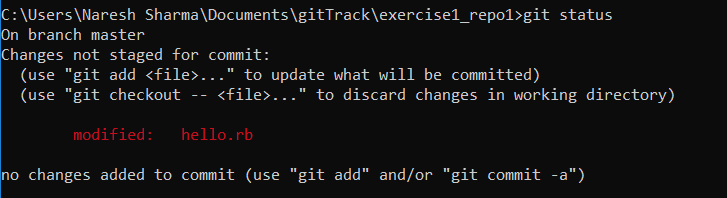


Command Used : - git reset –hard <commit-hash>

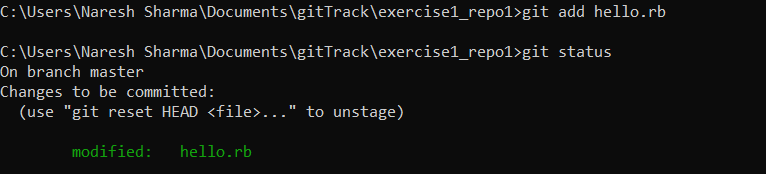


1. Check repository status before adding changes to staging area, after adding changes to staging area and after committing changes .

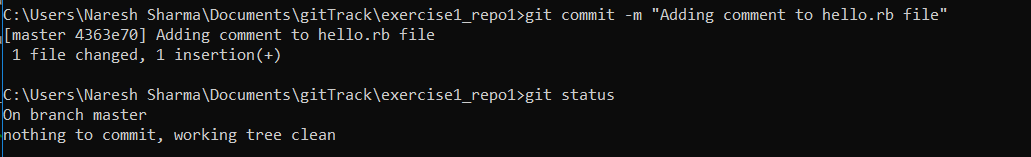
before adding changes to staging area



after adding changes to staging area

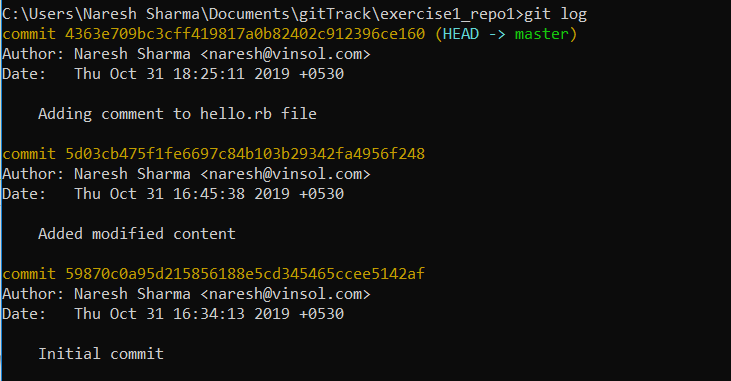


after committing changes



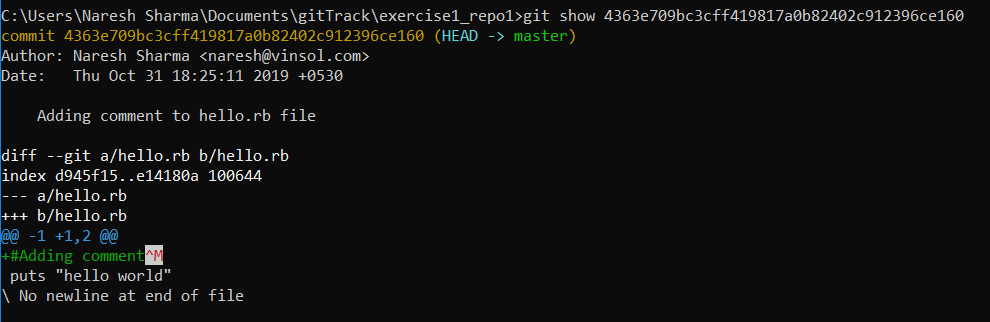
1. Display all the commits .

Command :- git log

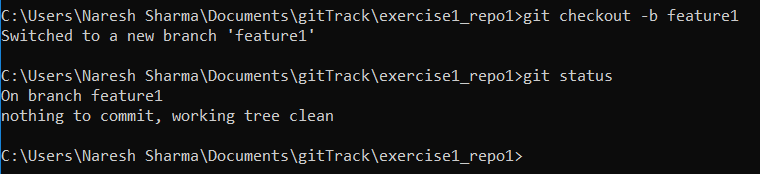


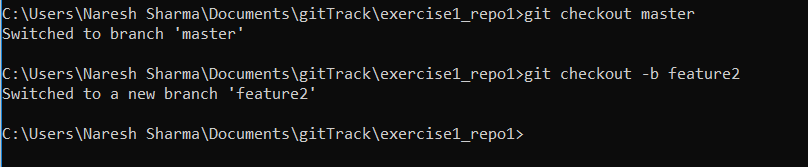
1. Display a specific commit .

Command :- git show <commit-hash>

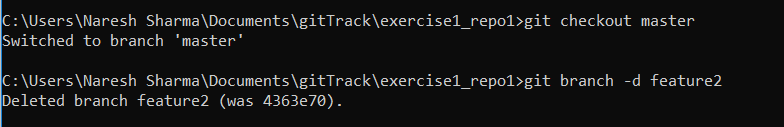


1. create branches called "feature1" and "feature2" from master.

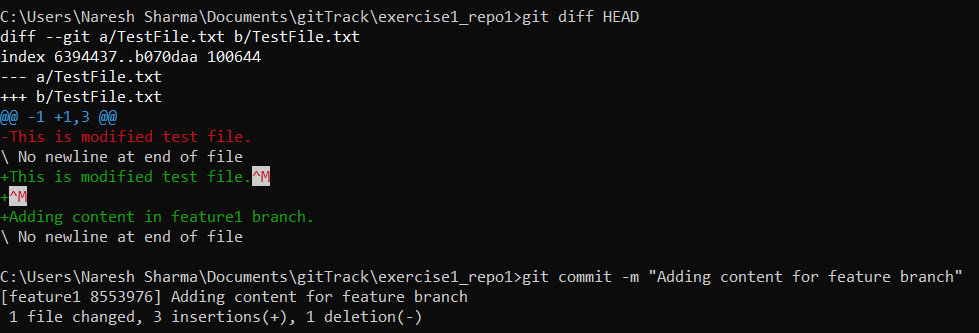




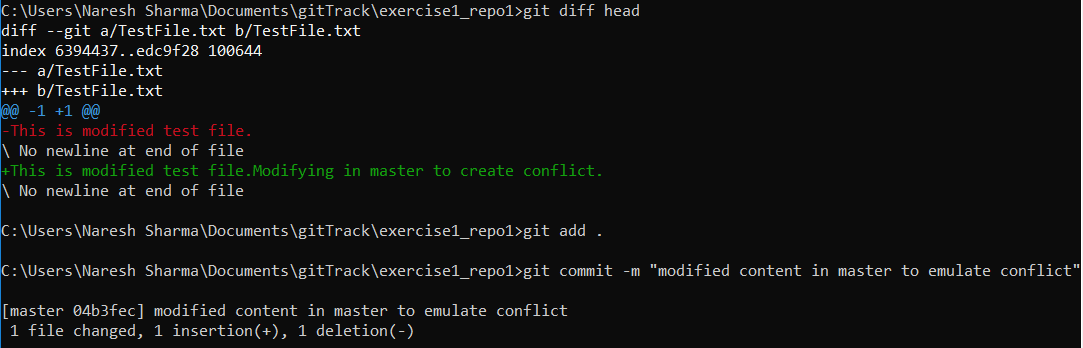
Delete branch "feature2".



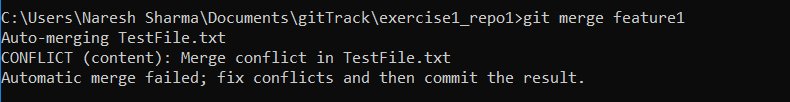
1. Make some commits on "feature1".

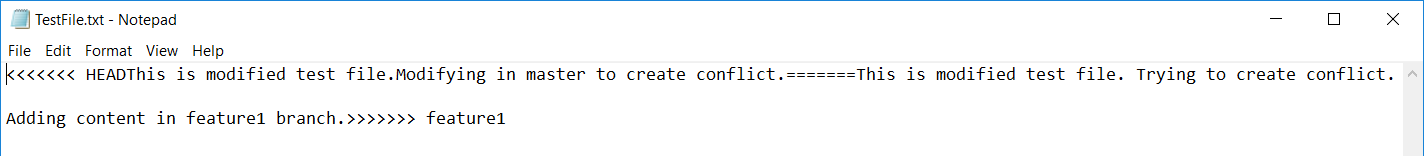


1. Make some commits on branch "master".

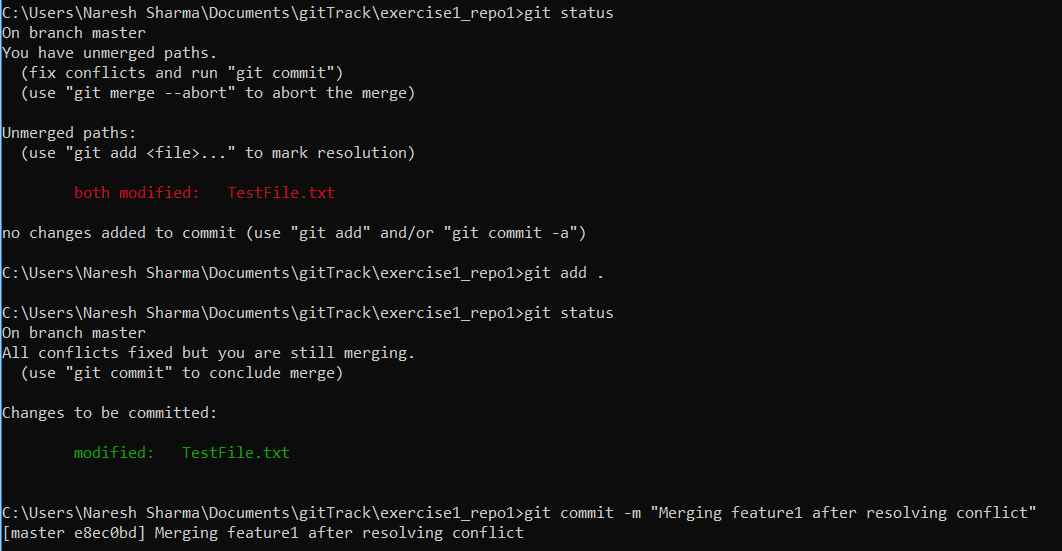


1. merge branch "feature1" in "master" (make some conflicts by changing same file on same line in both the branches)

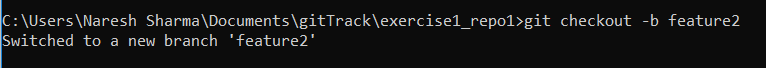




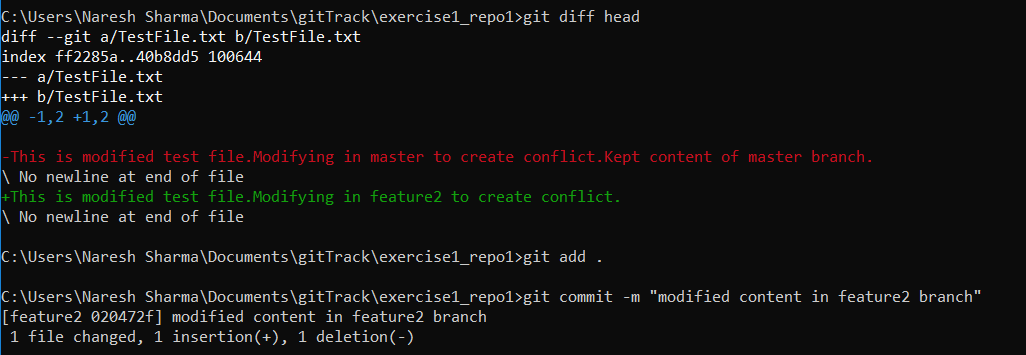
Resolved conflict and then added the file to staging area and then committed which completed the merge operation.

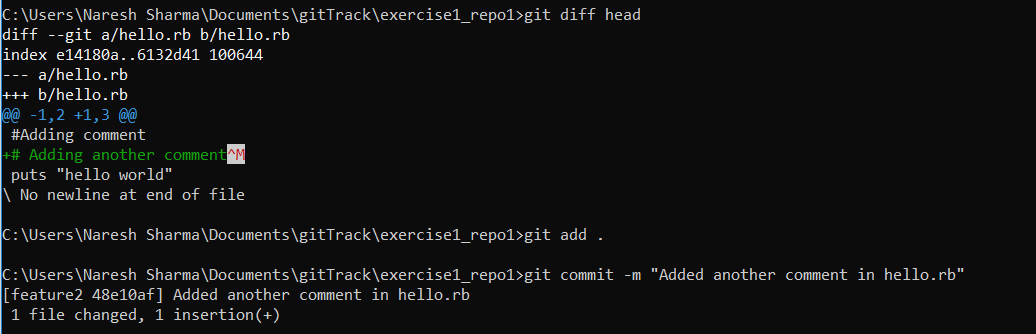


create branch "feature2" from master .

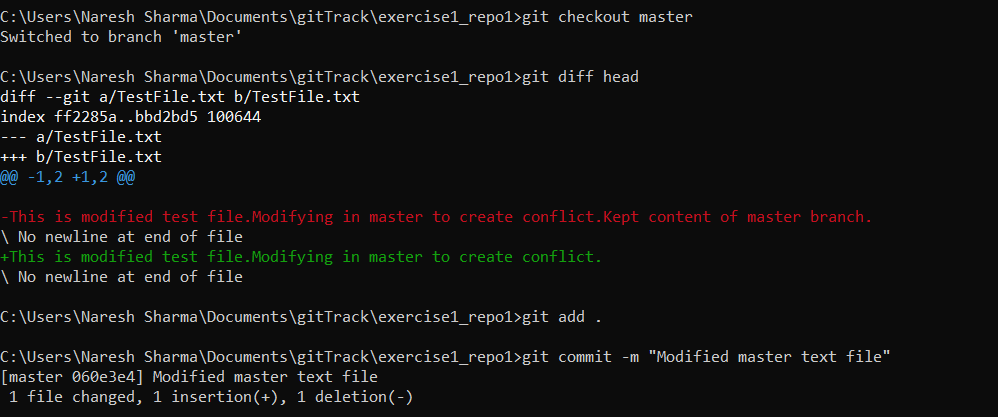


1. Make some commits on "feature2".

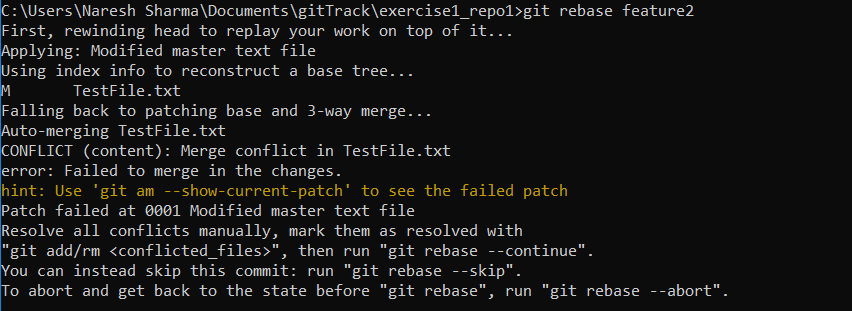




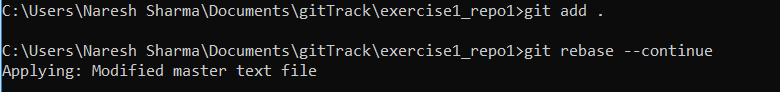
1. Make some commits on branch "master".



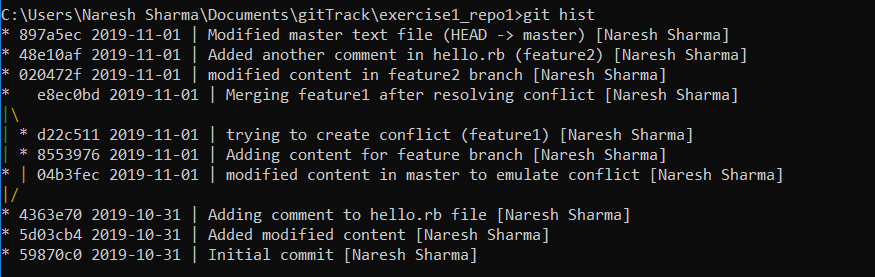
1. rebase branch "master" in "feature2" (make some conflicts by changing same file on same line in both the branches) .

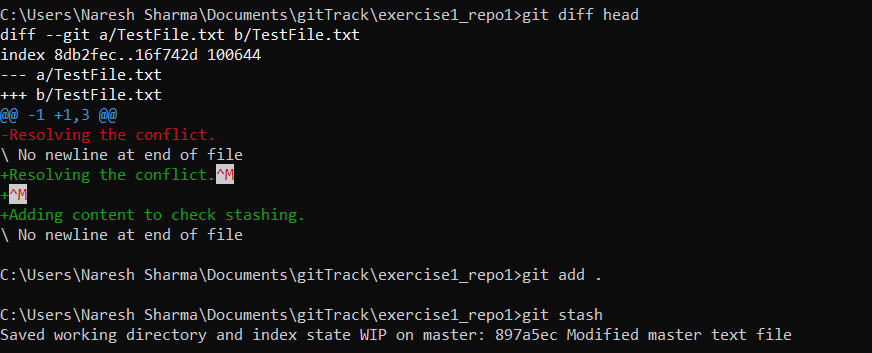


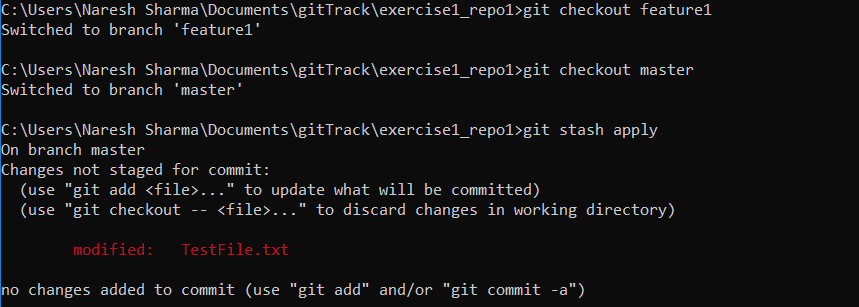
Modified conflicting file and resolved conflict. Then used git rebase --continue to complete the merge operation

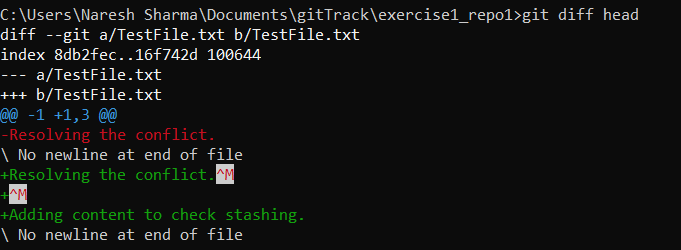


1. find the differences between "merge" and "rebase" in above cases.



1. Use git stash





1. create a repository on github for your working repository and push changes on remote repository.

