1. **Set your username and email in git config**

   git config --global user.name "KALLURINAVEENKUMAR"
   git config --global user.email "naveen.kalluri@practo.com”

2. **Create a new branch named "feature-branch" and switch to it.**

   git branch feature-branch
   git checkout feature-branch

3. **List all branches in the repository.**

   git branch -a

4. **Delete the branch “feature-branch”**

   git branch -d feature-branch

5. **How do you undo the last commit**

   git revert HEAD

6. **Create a new branch names “conflict-branch”**

   git branch conflict-branch

7. **Create a another branch named “feature1”**

   git branch feature1

8. **Make some changes in to feature1 branch**

   git switch -c feature1

9. **Merge “feature1” branch into main branch**

   git checkout main

   git merge feature1

10. **Make changes in “conflict-branch”, in the same file and line that you had made changes in feature1**

    git checkout conflict-branch

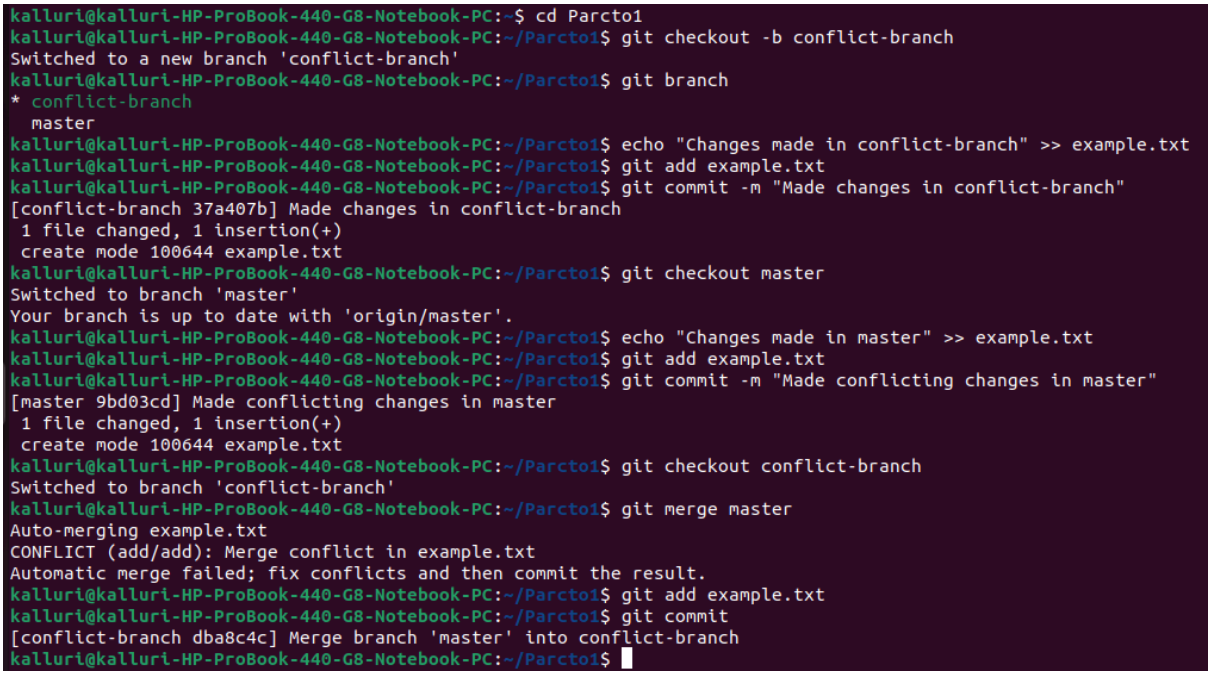
    git switch conflict-branch

    git add example.txt

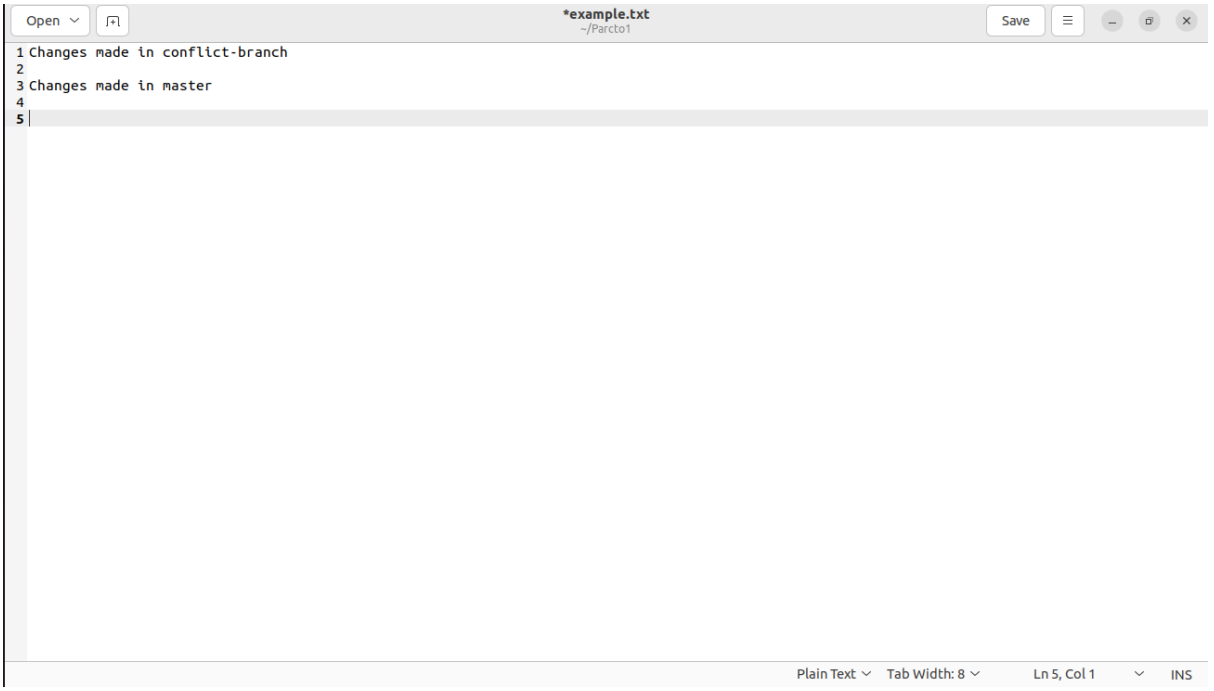
git commit -m "Made conflicting changes in conflict-branch"

git checkout main
git merge conflict-branch

11. Merge master into conflict-branch [Attach screenshot of terminal & file]



Example.txt



12. Resolve merge conflicts

First I will use git status to see the status of my repo and next to resolve the conflicts which I faced for the above one , where git inserts conflict markers , I will remove those markers.

For example:

For above question i got this first

<<<<<<< HEAD
Changes made in conflict-branch
=======
Changes made in master
>>>>>>> master

Then i removed those conflict markes (<<<<<<< , ======= , >>>>>>>)

Issue Solved.

13. Add a remote named "origin" pointing to a GitHub repository.

git remote add origin https://github.com/your-username/your-repository.git

14. **Fork a repository on GitHub and clone it to your local machine.**

    Add another repo and adding into existing repo , by this existing code will not get disturbed for that ,

    git remote add upstream https://github.com/anotherrepo/anotherrepo.git
    git fetch upstream
    git merge upstream/main

15. **Create a new branch on your fork, make changes, and open a pull request to the original repository.**

    git clone https://github.com/KALLURINAVEENKUMAR/practo2

    cd practo2

    git checkout -b newbranch

    vim file.txt (“first content”)

    git add file.txt

    git commit -m "Updated content"

16. **Comment on a PR and suggest improvements**

    Great work on this PR! The new feature adds valuable functionality. However, I noticed a potential performance bottleneck in the addition( ) function. Consider optimizing it for better scalability.

17. **Create a Git alias for the command `git log --oneline` named `gitlol`.**

    [alias]
       gitlol = log --oneline

18. **Create a pre-commit hook**

    In the hooks directory in our repo we can create a Pre-Commit Hook and can add our desired commands . when each commit runs the script or commands written in the Pre-Commit hook automatically runs.

19. **you have made local changes in your branch, but you need to switch to another branch urgently without committing. How would you handle this situation?**

    I can create another branch using

git checkout other-branch

20. **You accidentally deleted a file in your local repository. How do you restore it using Git?**

    First by checking git log to find the commit where the file was deleted by using

    git log - - path_to_deleted_file

    Next i will restore the deleted file using git checkout

21. **You have committed changes to your branch but forgot to include a file. How do you add the file to the last commit without creating a new commit?**

    First i will add foregetten file using

    git add (path of forgotten file)

    Next i will use - - amend and - -no-edit

    Because - - amend allows us to modify the last commit and - -no-edit allows that the commit message of the last commit remains unchanged.

22. **You want to discard all changes in your working directory and revert to the last commit. What Git command would you use?**

    git reset –hard HEAD

23. **You need to view specific commit changes. What Git command can be used to show the changes introduced by a particular commit?**

    git show

24. **You want to change a commit message, after you have already committed, how do you do so?**

    git commit --amend . This allows me to change the most recent commit message in your branch.

25. **Your colleague has made changes in their branch, and you want to incorporate those changes into your branch without merging. How do you achieve this?**

    git cherry-pick - By this I can selectively apply changes from your colleague's branch onto your own branch without merging the entire branch history.

26. **You've made several commits on a branch, but you want to club them into a single commit before pushing to the remote repository. How would you do that?**

    git rebase -i , This allows me to efficiently combine multiple commits into a single commit before pushing to the remote repository.

27. **You accidentally staged a file that you don't want to commit. How do you unstage it?**

    git reset HEAD pathtofile , this allows me to o unstage a file that was accidentally added to the staging area.

28. **You don’t want to commit files that have .yml in the end, and also files inside folder config. How do you do that?**

    In the .gitignore file I will add all .yml files.

29. **You want to see a list of all the files changed in the last commit. What Git command would you use?**

    git show --name-only HEAD, This allows me to view list of all the files changed in the last commit.

30. **You realize that your local branch is outdated, and you want to fetch the latest changes from the remote repository. How do you do this without merging?**

    git fetch
    git rebase

31. **You accidentally deleted a branch. How do you recover it?**

    git reflog - used to find the commit where the branch was deleted

git checkout -b - recreate it locally

32. **You want to remove untracked files and directories from your working directory. What Git command would you use?**

    git clean

33. **You have a commit from a feature branch that you want to apply to the main branch without merging the entire feature branch.**

    Using **git cherry-pick** we can apply to the main branch without merging the entire feature branch.

34. **You mistakenly committed a change to the wrong branch and need to apply that commit to the correct branch.**

    Using, **git cherry-pick ,** we can selectively apply a commit from one branch where it was mistakenly committed.

35. **There is a series of commits on a feature branch, but you only want to cherry-pick a specific range of commits.**

    git cherry-pick <start-commit>^ ..<end-commit>

36. **You want to clone a GitHub repository onto your local machine, but you only need a specific branch. How can you achieve this?**

    git clone -b branch-name URL

37. **You've made changes to your local repository and want to push them to your fork on GitHub. What Git commands would you use?**

    git add .

    git commit -m "commit message"

git push origin branch-name

38. **You want to create a new branch both locally and on GitHub to work on a new feature. What commands would you use?**

    git checkout -b new-feature-branch
    git push -u origin new-feature-branch

39. **You want to see the commit history of a GitHub repository. How can you do this using Git commands?**

    git log is used to see the commit history of a GitHub repository.

40. **You've accidentally committed sensitive information and want to remove the commit from both your local and remote repositories on GitHub. What commands would you use?**

    git rebase -i HEAD~n and then delete the commit line and then use this command

    git push --force-with-lease origin branch-name

41. **You want to delete a remote branch on GitHub. What Git command would you use?**

    git push origin --delete branch-name

42. **Create a git repository for all your assignments and upload them in it. Ask your peers to code review it, and you need to code review your peers assignments**

43. **Create a pull request on any open source library on github, attach the pull request link to the readme file of this project's repository**