**GIT ASSIGNMAENT**

**Question 1**

Step 1: Create a new Git repository.

A screenshot of a computer

Description automatically generated

Step 2: Create a file and commit changes.

A screenshot of a computer screen

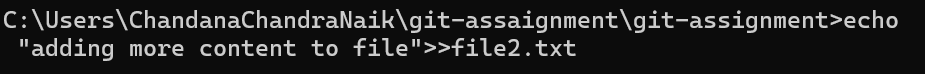
Description automatically generated

Step 3: View the commit history of your repository.

A screenshot of a computer

Description automatically generated

Step 4: Open the file you created earlier and make some changes to it.



Step 5: Check the file you modified is now marked as "modified" and unstaged.

Hint (git status)

A screen shot of a computer

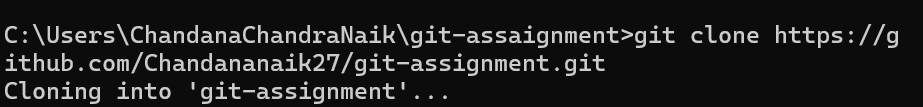
Description automatically generated

Step 6: Stage the changes you made to the file and commit the changes to the repository.

A screenshot of a computer screen

Description automatically generated

Step 7: Clone the repository you have created in GitHub.



Step 8: Fetch the changes, navigate into the cloned repository using the command line, and use the command git fetch to fetch any changes that have been made to the original repository since you cloned it.

A screen shot of a computer

Description automatically generated

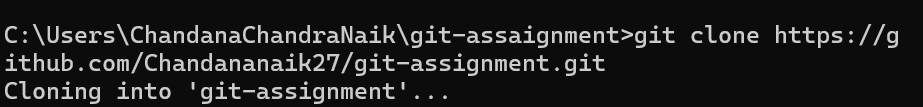
Step 9: Pull changes, merge the changes you just fetched into your local copy of the repository, and use the command git pull.

A black screen with white text

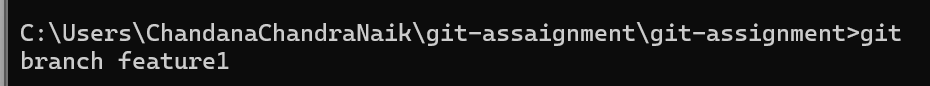
Description automatically generated

**Question 2**

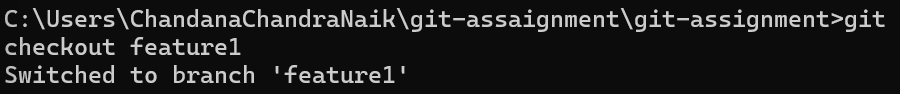
Step 1: Clone the repository you have created in GitHub.



Step 2: Create a new branch using the command.



Step 3: Switch to the new branch.



Step 4: Make some changes to the code in your local copy of the repository.



Step 5: Commit changes to the new branch.

A screenshot of a computer screen

Description automatically generated

Step 6: Switch back to the original branch

A black background with white text

Description automatically generated

Step 7: Merge the new branch.

A screen shot of a computer

Description automatically generated

Step 8: Push changes to the original branch

A screenshot of a computer program

Description automatically generated

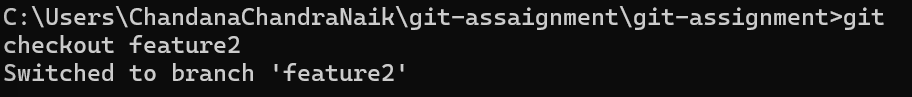
**Question 3**

Step 1: Create a feature branch.

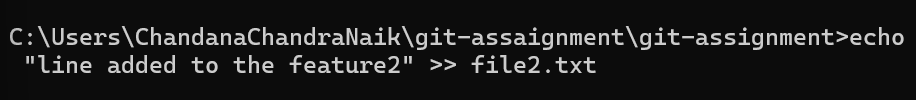
A screen shot of a computer

Description automatically generated

Step 2: Switch to the new branch.



Step 3: open the file and make some changes to it.



Step 4: Add and commit the changes to the new branch.

A screenshot of a computer program

Description automatically generated

Step 5: Push the changes to the new feature branch.

A screenshot of a computer screen

Description automatically generated

Step 6: Create a pull request.

A screenshot of a computer

Description automatically generated

Step 6: As another user in the master branch make some changes to the same file.

A screen shot of a computer

Description automatically generated

Step 7: Add and commit the changes to the master branch.

A screenshot of a computer screen

Description automatically generated

Step 8: Push the changes to the master branch.

A screen shot of a computer program

Description automatically generated

Note: There will be a conflict in the pull request, how do we resolve it??

Hint: git rebase

A screenshot of a computer program

Description automatically generated

A screenshot of a chat

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer email

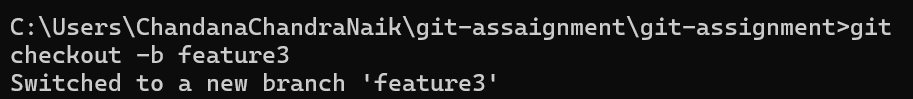
Description automatically generated**

**A screenshot of a computer screen

Description automatically generated**

**Question 4**

Step 1: Step 1: Create a feature branch.



Step 2: Switch to the new branch.

1)Open the file and make some changes to it and add and commit the changes to the new branch.

A screenshot of a computer program

Description automatically generated

2)Open the same file and make some changes to it and add and commit the changes to the new branch.

A screenshot of a computer screen

Description automatically generated

3)Open the same file and make some changes to it and add and commit the changes to the new branch.

A screenshot of a computer screen

Description automatically generated

Step 3: Identify the commit or commits that you want to "cherry-pick"(Note the hash of the commit or commits that you want to "cherry-pick")

A screen shot of a computer program

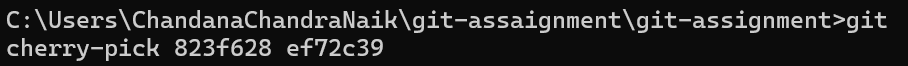
Description automatically generated

Step 4: Use the "git checkout" commagitnd to switch to the branch where you want to apply the changes.

A black background with white text

Description automatically generated

Step 5: Use the "git cherry-pick" command followed by the commit hash(es) that you want to apply.



Before Cherry-pick:

A screenshot of a computer

Description automatically generated

After Cherry-pick:

A screenshot of a computer

Description automatically generated

Conflicts:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

**Question 5**

Step 1: Step 1: Create a feature branch.

A black background with white text

Description automatically generated

Step 2: Switch to the new branch.

1)Open the file and make some changes to it and add and commit the changes to the new branch.

A screenshot of a computer program

Description automatically generated

2)Open the file and make some changes to it and add and commit the changes to the new branch.

A screen shot of a computer

Description automatically generated

3)Open the file and make some changes to it and add and commit the changes to the new branch.

A screenshot of a computer screen

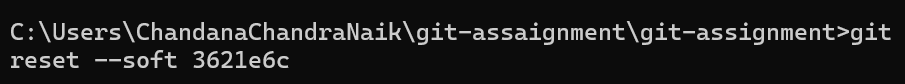
Description automatically generated

Step 3: Use the "git log" command to view the commit history and identify the commit to which you want to reset.

A screen shot of a computer

Description automatically generated

Step 4: Use the "git reset" command followed by the desired reset type and the commit hash





A black screen with white text

Description automatically generated

Step 5: Verify that the reset was successful by using the "git log" command again.

A computer screen with text

Description automatically generated

Step 6: Use the "git log" command to view the commit history and identify the commit that you want to reverse.

A screen shot of a computer

Description automatically generated

Step 7: Use the "git revert" command followed by the commit hash or reference to which you want to revert. (Hint: git revert <commit hash>)

A computer screen shot of a black screen

Description automatically generated

Step 8: Verify that the revert was successful by using the "git log" command again.

Note: Identify the difference between git log after git reset and git r evert.

A screen shot of a computer program

Description automatically generated