

PROGRAM 3

Write the commands to stash your changes, switch branches, and then apply the stashed changes.

Let's break down each command step-by-step:

STEP 1: `git branch new:`

- This command creates a new branch named "new". This new branch is a copy of the current branch's state.

STEP 2: `git checkout new:`

- This command switches your working directory to the newly created "new" branch. Any changes you make from now on will be specific to this branch.

STEP 3: `vi file4.txt:`

- This command opens the file "file4.txt" in the vi text editor. You can edit the contents of this file.

STEP 4: `git stash:`

- This command temporarily saves your uncommitted changes. This is useful when you need to switch branches or perform other tasks without committing your current work.

STEP 5: `git checkout master:`

- This command switches your working directory back to the "master" branch.

STEP 6: `git stash apply:`

- This command applies the saved changes from the stash to the current branch, which is "master" in this case.

STEP 7: `git status:`

- This command shows the current state of the working directory. It will show that there are changes to be committed.

STEP 8: `git add .`:

- This command stages all the changes made to tracked files in the current directory, which are the changes applied from the stash.

STEP 9: `git status`:

- This command shows the current state of the working directory. It will show that there are staged changes to be committed.

STEP 10: `git commit -m "stash changed"`:

- This command creates a new commit on the "master" branch with the message "stash changed". The staged changes are included in this commit.

STEP 11: `git log`:

- This command shows the commit history of the current branch. You should see the commit you just made, which includes the changes from the stash. In essence, what we've done is:
- Created a new branch to work on a specific feature or bug fix.
- Made changes to the file "file4.txt" on the new branch.
- Temporarily saved the changes using `git stash`.
- Switched back to the main branch.
- Applied the saved changes to the main branch using `git stash apply`.
- Committed the applied changes to the main branch.

This workflow is useful when you need to switch between tasks or branches without losing your current work.

By stashing your changes, you can temporarily save them and restore them later when needed.