**Day6 – Assignment3:** Write a step to create a new branch and merge with master branch.

1. **Ensure you are on the master branch:**

Bash

* git checkout master

1. **Update master branch (optional, to ensure you have the latest changes):**

Bash

* git pull origin master

1. **Create a new branch:**

Replace <branch-name> with your desired branch name.

Bash

* git checkout -b <branch-name>

This command creates a new branch (<branch-name>) and switches you to it (-b flag combines git branch and git checkout commands).

1. **Make changes and commit them:**

Make your changes to the files in your working directory (workspace).

Bash

# Make changes to files

# Stage changes

* git add .

# Commit changes

* git commit -m "Your commit message here"

1. **Switch back to the master branch:**

Bash

* git checkout master

1. **Merge the new branch into master:**

Bash

* git merge <branch-name>

This command integrates the changes from <branch-name> into the current branch (master in this case).

1. **Resolve any merge conflicts (if applicable):**

If Git detects conflicting changes between the branches, you'll need to resolve them manually. Git will mark the conflicted areas in the affected files. After resolving conflicts, stage the resolved files

(git add) and commit the merge (git commit).

1. **Push changes to remote repository (if necessary):**

If you want to push the merged changes to a remote repository (e.g., GitHub, GitLab):

Bash

* git push origin master

Notes:

• Replace <branch-name> with a meaningful name for your branch.

• Ensure you commit your changes before merging.

• Handle merge conflicts carefully by resolving them and committing the changes.

• Always communicate with your team if you're making changes to shared branches like master.

By following these steps, we can create a new branch, make changes, merge them into the master branch, and effectively manage your Git workflow.