PROGRAM 6

Write the command to merge "feature-branch" into "master" while providing a custom commit message for the merge.

STEP 1: git branch feature-branch:

* Purpose: Creates a new branch named "feature-branch" off of the currently checked-out branch.

* How it works:

* Git creates a new pointer to the current commit.

* This new pointer is named "feature-branch."

* You can now make changes on this new branch without affecting the original branch.

STEP 2: git checkout feature-branch:

* Purpose: Switches the active branch to "sub-branch."

* How it works:

* Git updates the working directory to match the state of the "sub-branch."

* Any changes you make from this point on will be applied to the "sub-branch."

STEP 3: vi file5.txt:

* Purpose: Opens the file "file5.txt" in the vi text editor.

* How it works:

* This command is not specific to Git but is a common Linux command toedit text files.

* You can use vi to make changes to the file, which will be reflected in the "sub-branch" when you commit.

STEP 4: git commit -m "commit":

Commits the staged changes with the message "commit".

STEP 5: git checkout master:

Switches to the master branch (or main, depending on your project setup).

STEP 6: git merge feature-branch:

Merges the changes from feature-branch into the current branch (master in this case).

STEP 7: git log:

* Purpose: Shows the commit history of the current branch.
* How it works:
* Git displays a list of commits, starting from the most recent.
* Each commit is shown with its hash, author, date, and a brief message.
* This command helps you understand the history of your project andidentify specific changes.

OUTPUT:

```
MINGW64 ~/061/clone/gitclone (main)
$ git branch feature-branch

MINGW64 ~/061/clone/gitclone (main)
$ git checkout feature-branch
Switched to branch 'feature-branch'

MINGW64 ~/061/clone/gitclone (feature-branch)
$ vi file5.txt

MINGW64 ~/061/clone/gitclone (feature-branch)
$ git commit -m "Commit"
On branch feature-branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file5.txt

nothing added to commit but untracked files present (use "git add" to track)

MINGW64 ~/061/clone/gitclone (feature-branch)
$ git add .
warning: in the working copy of 'file5.txt', LF will be replaced by CRLF the next time Git touches it

MINGW64 ~/061/clone/gitclone (feature-branch)
$ git log
commit 9567e815b567cfa01486d83fee4b150038d4bd63 (HEAD -> feature-branch, origin/main, origin/HEAD, main)
Author: Vinayaka N Walishettar# <walishettar123@gmail.com>
Date:   Sun Dec 22 10:02:07 2024 +0530

    Create git061
```