Git Branching and Merging

In this lab, you will create, merge and delete git branches.

Part 1: Branching

1. Verify you are on the master branch (noted by an asterisk)
   $ git branch

```
/home/ubuntu/my-repo --> git branch
* master
```

2. Create and checkout a new branch named “testing”
   $ git checkout –b testing

```
/home/ubuntu/my-repo --> git checkout -b testing
Switched to a new branch 'testing'
```

3. Verify you are on the new branch
   $ git branch

```
/home/ubuntu/my-repo --> git branch
  master
* testing
```

4. Switch back to the master branch
   $ git checkout master

```
/home/ubuntu/my-repo --> git checkout master
Switched to branch 'master'
/home/ubuntu/my-repo -->
```

5. Verify you are on the master branch (noted by an asterisk)
   $ git branch

```
ubuntu@ip-172-31-9-70:~/my-repo$ git branch
* master
  testing
```

6. Switch back to the testing branch
   $ git switch testing

7. Verify you are on the testing branch
   $ git branch

8. Create a file named “file1.txt” with some content

9. Add file1.txt to the staging area of the testing branch
   $ git add file1.txt

10. Commit file1.txt. **Notice the branch name “testing” in the commit comments**

11. Look at the commit logs to see your commit message, including your newly added commit:
    $ git log

    a. Use ‘q’ to quit (if needed to exit from the 'git log' output)

12. Look at the files in your directory. You should see both README and file1.txt
    $ ls -la

13. Switch back to the master branch

14. Look at the files in your directory. **You should see only the 'README.md' file.** Why?
    $ ls -la

15. Switch between master and testing branches a few times and watch the existence of file1.txt change, because it only “belongs” to the testing branch

Part 2: Merging

16. Make sure you are on the master branch

17. Note the file contents of master branch only contains README and .git:
    $ ls -la
    **The command 'ls -la' means:
        - list the contents of the current directory
        - show the long listing for each item (-l)
        - show all hidden files and directories (-a). Hidden files and directories in Linux start with a dot, such as '.git'

18. Merge the testing branch into the current master branch
    $ git merge testing

```
/home/ubuntu/my-repo --> git merge testing
Updating 137cec5..556c960
Fast-forward
 file1.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

19. Read the message showing the changes

20. The master branch contents now contain:
    file1.txt, README and .git

21. Compare contents of both branches, note they are now identical. Using 'git checkout' might be useful for this.

Part 3: Deleting

22. Create a new branch, which we can delete, named “newbranch”
    $ git checkout -b newbranch

23. Verify you are on the new branch
    Note the list of branches includes:
    master, newbranch and testing

24. Delete the ‘newbranch’ branch
    $ git branch -D newbranch
    Notice you can’t delete the current branch. You have to first checkout a different branch in order to delete the current branch

25. Change to the testing branch, then delete ‘newbranch’

26. Verify your current branch. Note the list of all branches no longer includes ‘newbranch’

Notify your instructor that you are done with the lab

END OF LAB