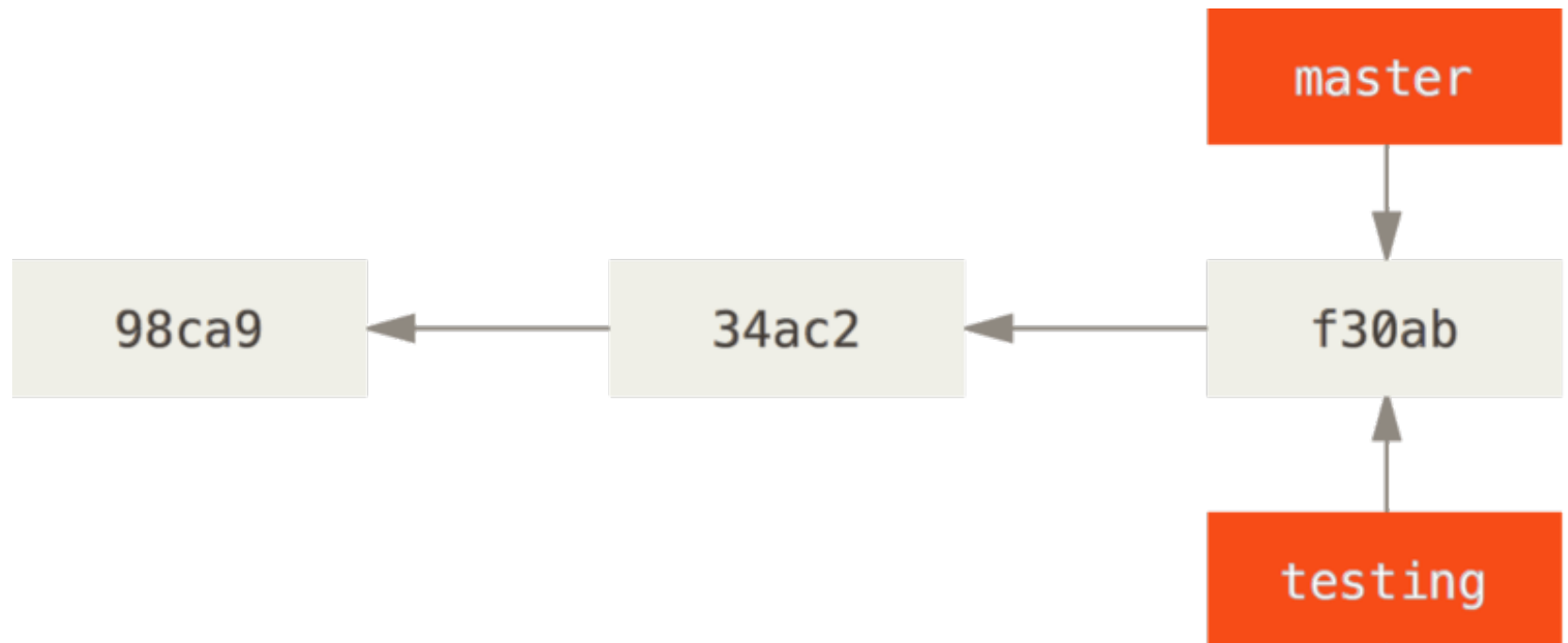


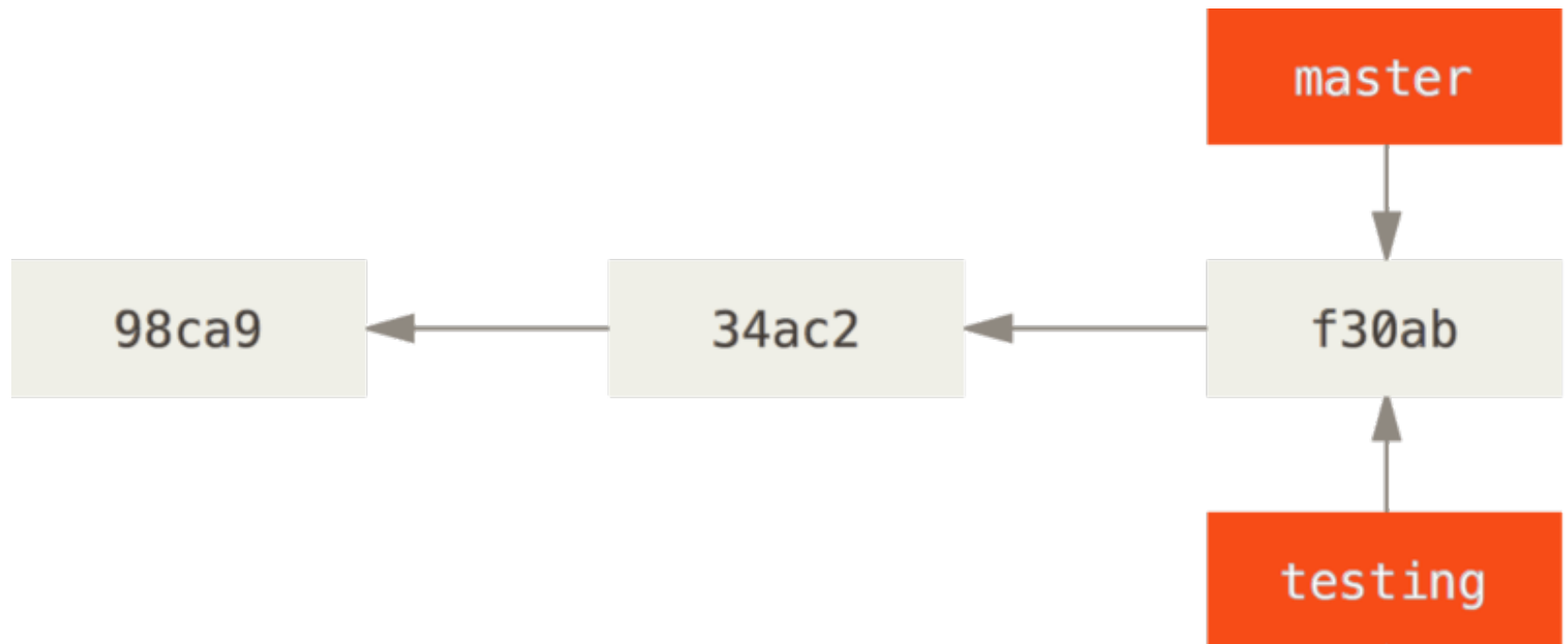
# LABWORK: GIT BRANCHING AND MERGING

- ▶ In Git Bash (or terminal) navigate so that you're inside the 2240Lab2 repository from last week.
- ▶ If you don't have a repository from last week, go to GitHub, create a repository called 2240Lab2, then clone it locally
- ▶ You can create a new branch in git using the branch command.
- ▶ Create a new branch called *testing*:  
*git branch testing\_yourFirstName\_LastName*
- ▶ **Do a *git status* to show your new branch and take a screenshot**



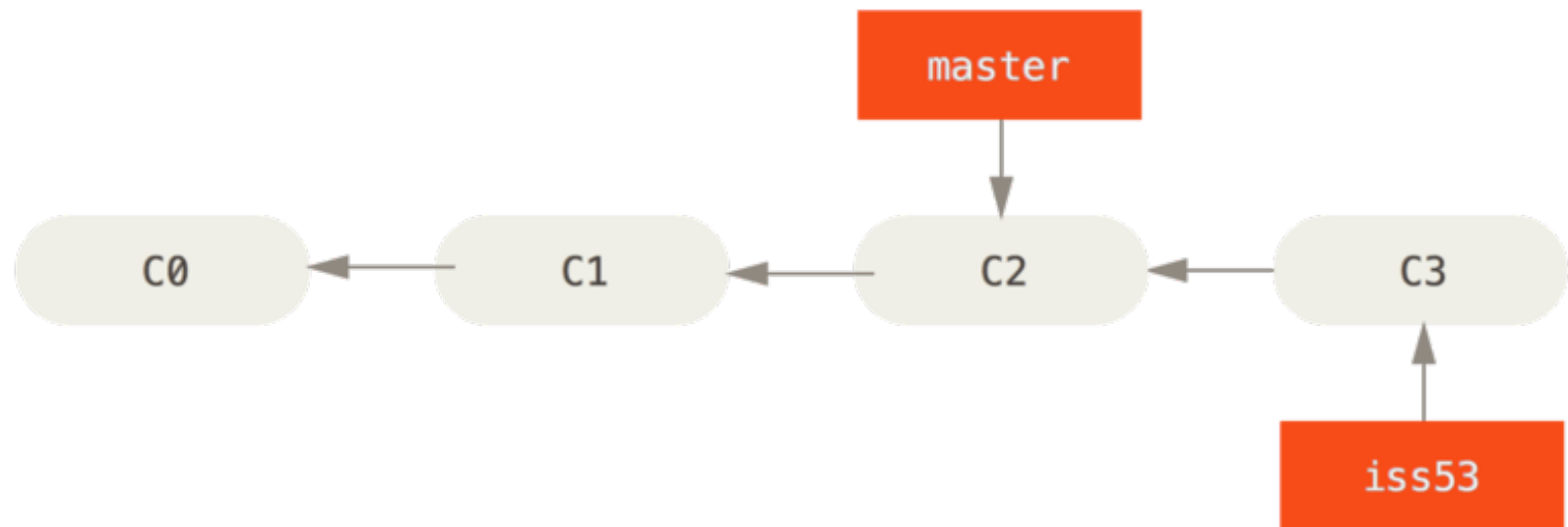
# LABWORK: GIT BRANCHING AND MERGING

- ▶ Now switch to testing: *git checkout testing\_yourFirstName\_LastName*
- ▶ Create a new file called testing.txt in the new branch and then add and commit it.
- ▶ Note: in your local file system the master and all branches are represented by the same directory
- ▶ **Print the log history of your branch and screenshot it**



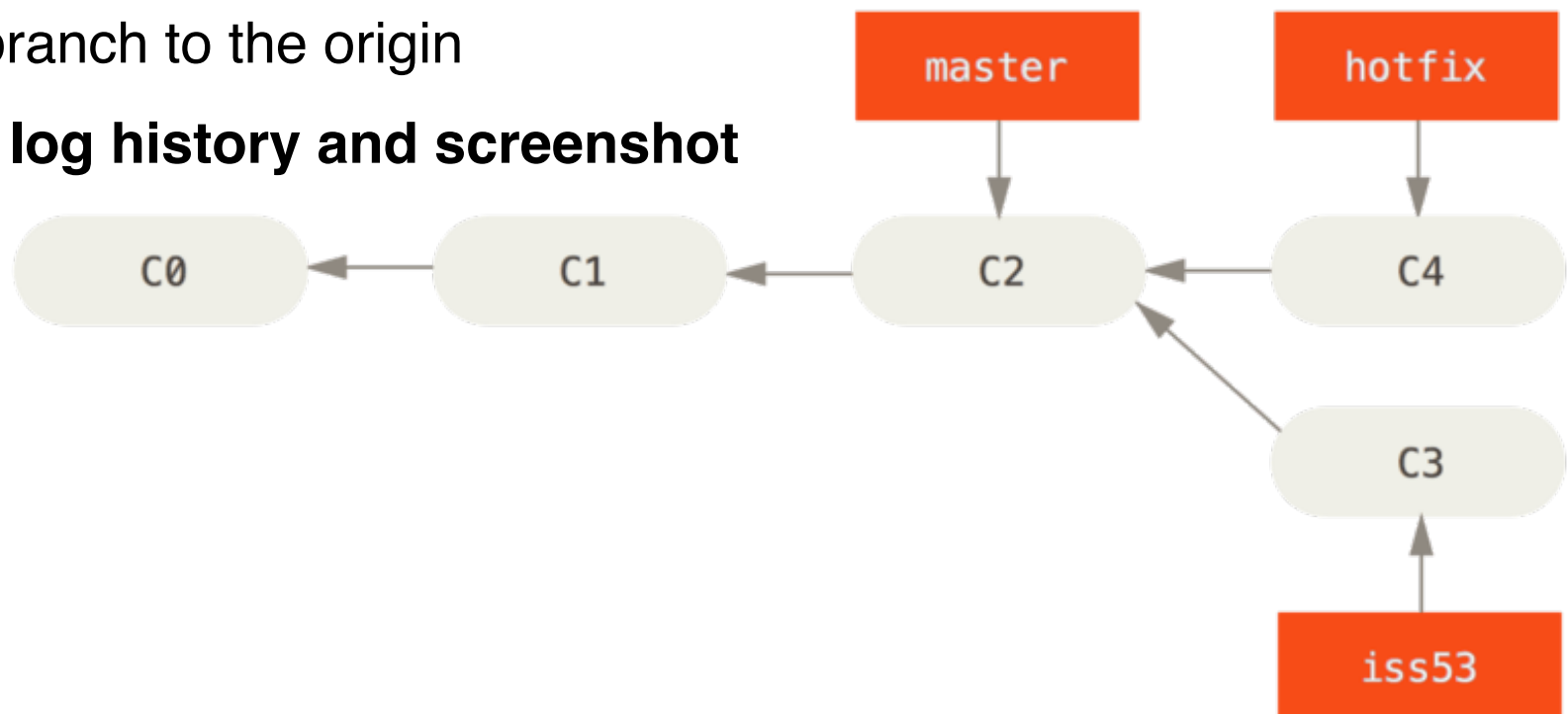
# LABWORK: GIT BRANCHING AND MERGING

- ▶ Push the new branch to remote
- ▶ *git push -u origin testing\_yourFirstName\_LastName*
- ▶ The -u automatically sets upstream for you, linking your local repo to a central one.
- ▶ "Upstream" would refer to the remote repo that other people will be pulling from, e.g. your GitHub repo.



# LABWORK: GIT BRANCHING AND MERGING

- ▶ Now create another branch called *hotfix\_yourFirstName\_LastName* using the command:
- ▶ *git checkout -b hotfix\_yourFirstName\_LastName*
  - ▶ Note this does both the branching and checkout in one step
- ▶ Create a new file called *hotfix.txt* and then add and commit to *hotfix\_yourFirstName\_LastName*
- ▶ Push this branch to the origin
- ▶ **Print your log history and screenshot**



# LABWORK: GIT BRANCHING AND MERGING

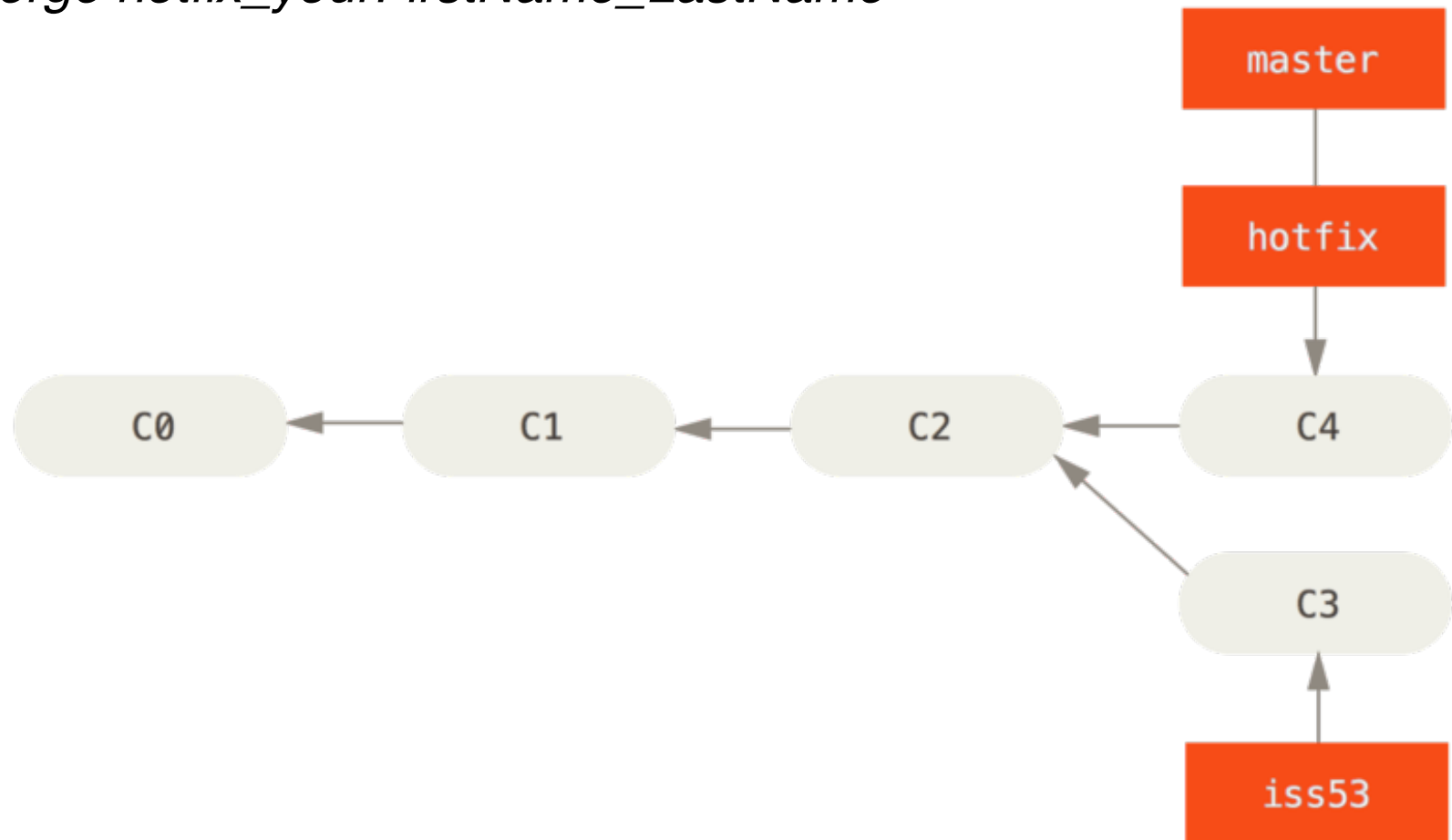
- ▶ Go to GitHub and notice the difference between the three branches you now have.
- ▶ Why does the hot fix branch have the testing.txt file? How would you avoid this?
- ▶ **Include a screenshot of your github, showing the hot fix branch**

# LABWORK: GIT BRANCHING AND MERGING

- ▶ We will merge hotfix\_yourFirstName\_LastName with master

*git checkout master*

*git merge hotfix\_yourFirstName\_LastName*



# LABWORK: GIT BRANCHING AND MERGING

- ▶ Now delete *hotfix\_yourFirstName\_LastName*

```
git branch -d hotfix_yourFirstName_LastName
```

- ▶ Then delete the remote branch

```
git push origin --delete hotfix_yourFirstName_LastName
```

**Print and screenshot your log history**

# LABWORK: GIT BRANCHING AND MERGING

- ▶ Suppose we had changed a file in *hotfix\_yourFirstName\_LastName* that existed in master. This will result in conflict.
- ▶ In cases of conflict, you can resolve the conflict in the file by editing it. Then you can stage it by using the command `add`, and then `commit`.



# LABWORK: GIT BRANCHING AND MERGING

- ▶ Git is very interesting, right??
- ▶ You can learn more about git from the following resources:

<https://git-scm.com/> (an excellent resource)

J. Loeliger. *Version Control with Git*. O'Reilly, Sebastopol, CA, May 2009.