A. A fast-forward merge

- ✓ In a previous lab, you should have created a commit in your projectc repository with a fileA.txt file containing a string "feature 1". The commit message should be "add feature 1". This commit should be on the master branch. If you do not have this commit, create it now.
- ✓ Create and checkout a branch off of the latest master commit named "feature2". Use the same process that you used in the previous lab.
- ✓ In your local repository, create a commit on the feature2 branch with the following:
  - modify fileA.txt , adding "feature 2" directly under the line "feature 1"
  - add a commit message of "add feature 2"
- ✓ Use git log --oneline --graph --all view your commit graph. You should see a straight line, with your feature2 branch label and "add feature 2" commit message on the most recent commit. You should see HEAD -> feature2 .
- ✓ Let's assume that feature 2 is ready to be merged into the master branch. Start by checking out the master branch.
- ✓ Execute git merge feature2 . By default, this command will perform a fast-forward merge if possible.
- ✓ You should now see a linear history with the master and feature2 labels on the most recent commit. The fast-forward merge simply moved the master branch label to the latest commit.
- ✓ Delete the feature2 branch label.

Congratulations, you have performed a fast-forward merge.

B. Perform a merge with a merge commit

- ✓ Repeat the process above to create a feature3 branch and commit.
- ✓ This time, when you are ready to merge in the feature3 branch, execute git merge --no-ff feature3 . The --no-ff option will create a merge commit, resulting in a non-linear history. Verify that a merge commit was created.
- ✓ Delete the feature3 branch label.
- ✓ You will not use the projectc repository in future labs. You can delete it.

Congratulations, you have performed a merge with a merge commit.