Solve merging problem - <https://stackoverflow.com/questions/15745045/how-do-i-resolve-git-saying-commit-your-changes-or-stash-them-before-you-can-me>

You can't merge with local modifications. Git protects you from losing potentially important changes.

You have three options:

* Commit the change using
* git commit -m "My message"
* Stash it.

Stashing acts as a stack, where you can push changes, and you pop them in reverse order.

To stash, type

git stash

Do the merge, and then pull the stash:

git stash pop

* Discard the local changes

using git reset --hard  
or git checkout -t -f remote/branch

Or: Discard local changes for a specific file

15

You can stash (save the changes in temporary box) then, back to master branch HEAD.

$ git add .

$ git stash

$ git checkout master

**Jump Over Commits Back and Forth:**

* Go to a specific commit-sha.
* $ git checkout <commit-sha>
* If you have uncommitted changes here then, you can checkout a new branch, Add, Commit, Push the current branch to the remote.
* # checkout a new branch, add, commit, push
* $ git checkout -b <branch-name>
* $ git add .
* $ git commit -m 'Changes in the commit'
* $ git push origin HEAD # push the current branch to remote
* $ git checkout master # back to master branch now
* If you have changes in the specific commit and don't want to keep the changes, you can do stash or reset then checkout to master (or, any other branch).
* # stash
* $ git add -A
* $ git stash
* $ git checkout master
* # reset
* $ git reset --hard HEAD
* $ git checkout master
* After checking out a specific commit if you have no uncommitted change then, just back to master or other branch.
* $ git status # see the changes
* $ git checkout master
* # or, shortcut

$ git checkout - # back to the previous state