# Worksheet 3 – Git and Branches

1. In your local *fall2019* repository, create a new branch called **unit2\_draft**.
2. See if the branch was created: you can use “git branch” command.
3. Switch to this new branch, create a new folder, called **Unit02** and inside of this folder, create a file **lecture02.txt**.
4. Add to the staging area and commit this new change.
5. Switch back to the master branch.
6. Make sure you are on master – type “git branch” and see if master has the “\*” symbol.
7. Merge **unit2\_draft** into the master branch.
8. Delete **unit2\_draft** branch.
9. Do “git push” to push to your remote repo.

Now, let’s try to create conflicting changes on both branches.

1. Create a branch called **unit3\_draft.**
2. Switch to this new branch, create a new folder, called **Unit03** and inside of this folder, create a file **lecture03.txt**.
3. Add to the staging area and commit this new change.
4. Still on **unit3\_draft** branch, open syllabus.txt and add some text, for example, “1. Use git.”
5. Save the file, add to the staging area and commit this new change.
6. Switch back to the master branch.
7. On the master branch, open syllabus.txt and add some text, for example, “1. Use GitHub.”
8. Save the file, add to the staging area and commit this new change.
9. Now, try merging **unit3\_draft** branch into master. (Check to make sure you are on master first!)
10. If you receive a merge conflict instead, solve it by selecting the final version of syllabus.txt that you want to keep. Add and commit this new change. The branches should then merge automatically.
11. Delete **unit3\_draft** branch.
12. Push the new changes to the remote.