

# CSc 210: Software Development

## Section 2: Git gud

Monday, September 11th, 2017

For this section, we will be practicing our git skills and how to deal with merge conflicts. We will also experiment a little with Java.

### Exercise: Using git with a partner

#### PLEASE READ THROUGH THE STEPS BEFORE WALKING THROUGH THEM

1. The first step of this section is to find a partner to work with. Maybe someone sitting next to you, hmm? You and your partner will be represented by two personas in this write up, Gold Leader and Red Leader. Decide who is who.
2. Next, you must both follow the invitation link below for this section's github assignment.  
<https://classroom.github.com/g/q2O9QkQa>
3. When you follow this link, you will need to accept the assignment. You have two tasks to complete: first Gold Leader will create a team for this assignment, then Red Leader will join that team. When you arrive at the page given above, you will see 1) a list of already created teams, and 2) an option to create a new team. As stated before, Gold Leader will create the team. Once that is done, Red Leader will join it. Please use the following format for naming your team name: *section\_number-Gold\_Leader\_netid-Red\_Leader\_netid*. The other team member must follow the link given above and join the same team. You will both now have access to a shared repository.
4. Now, you should both clone your repositories to your computers. Once done, you will each find a file called Hello.java in the local repos. You will both be adding code to this file. First, have Red Leader open the file and add code as instructed. Next, Red Leader should save the file, commit their changes, and push to github. Once Red Leader is finished, Gold Leader will pull the now updated file from github and make changes as instructed. Once done, Gold Leader should push their changes. After this, Red Leader will need to pull the changes Gold Leader just pushed.

5. Now, Hello.java should be in your repo updated with both Red and Gold Leader's print statements. At this point, you should both make sure to have updated repos on your computer's. Now, you should both add in the code in the callSign() method as it says at the same time.
6. Once each team member's code has been added, you should both commit your changes. Now, have Red Leader push their code to github. Then, have Gold Leader try to push their code. Notice how you are not allowed to do so: The code in the repository is "ahead" of the code on your machine. To fix this, try pulling the code again.
7. If done correctly, Gold Leader probably got an error message saying there was a merge conflict in Hello.java when they pulled. Why did this happen? Try opening that file and figuring out how to fix it (we want both of the print statements added to callSign() to remain in the file).
8. Once you have resolved the merge conflict, have Gold Leader push back up to github and have Red Leader pull the code so you have the same program on your computers. Compile Hello.java with javac and run it with the java command. You should both see the same output!

That's it! You have now been exposed to the workflow required to use github in a team environment.